WEST

Help | Logout | Interrupt

Main Menu | Search Form | Posting Counts | Show S Numbers | Edit S Numbers | Preferences | Cases

Search Results -

Terms	Documents
L1 same (type or class)	22

Database:
- US Patents Full-Text Database
- US Pre-Grant Publication Full-Text Database
- JPO Abstracts Database
- EPO Abstracts Database
- Derwent World Patents Index
- IBM Technical Disclosure Bulletins

Search: L2

Refine Search

Recall Text | Clear

Search History

DATE: Tuesday, August 12, 2003 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
	DB=USPT; PLUR=YES; OP=OR		
L2	L1 same (type or class)	22	L2
L1	detect$3 same dock$3 same bus	199	L1

END OF SEARCH HISTORY

WEST

Help | Logout | Interrupt

Main Menu | Search Form | Posting Counts | Show S Numbers | Edit S Numbers | Preferences | Cases

Search Results -

Terms	Documents
L2	0

Database:
US Patents Full-Text Database
US Pre-Grant Publication Full-Text Database
JPO Abstracts Database
EPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search: L3

Refine Search

Recall Text | Clear

Search History

DATE: Tuesday, August 12, 2003 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
DB=PGPB,JPAB,EPAB,DWPI,TDBD; PLUR=YES; OP=OR			
L3	L2	0	L3
DB=USPT; PLUR=YES; OP=OR			
L2	L1 same (type or class)	22	L2
L1	detect$3 same dock$3 same bus	199	L1

END OF SEARCH HISTORY

WEST

Help | Logout | Interrupt

Main Menu | Search Form | Posting Counts | Show S Numbers | Edit S Numbers | Preferences | Cases

Search Results -

Terms	Documents
l2 and L4	28

Database:
US Patents Full-Text Database
US Pre-Grant Publication Full-Text Database
JPO Abstracts Database
EPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search: L5

Refine Search

Recall Text | Clear

Search History

DATE: Monday, August 11, 2003 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
	DB=USPT; PLUR=YES; OP=OR		
L5	l2 and L4	28	L5
L4	L3 near10 dock$3	531	L4
L3	(obtain$3 or access$3 or read$3) near10 (device or module or unit)	461371	L3
L2	detect$3 same dock$3 same bus	199	L2
L1	detect$3 near10 dock$3 near10 bus	22	L1

END OF SEARCH HISTORY

WEST

Help Logout Interrupt

Main Menu | Search Form | Posting Counts | Show S Numbers | Edit S Numbers | Preferences | Cases

Search Results -

Terms	Documents
L5	0

Database:
US Patents Full-Text Database
US Pre-Grant Publication Full-Text Database
JPO Abstracts Database
EPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search: L6

Refine Search

Recall Text Clear

Search History

DATE: Monday, August 11, 2003 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
	DB=PGPB,JPAB,EPAB,DWPI,TDBD; PLUR=YES; OP=OR		
L6	L5	0	L6
	DB=USPT; PLUR=YES; OP=OR		
L5	l2 and L4	28	L5
L4	L3 near10 dock$3	531	L4
L3	(obtain$3 or access$3 or read$3) near10 (device or module or unit)	461371	L3
L2	detect$3 same dock$3 same bus	199	L2
L1	detect$3 near10 dock$3 near10 bus	22	L1

END OF SEARCH HISTORY

WEST

Help | Logout | Interrupt

Main Menu | Search Form | Posting Counts | Show S Numbers | Edit S Numbers | Preferences | Cases

Search Results -

Terms	Documents
((708/139)!.CCLS. |(361/683 |361/729 |361/686 |361/727)!.CCLS. |(709/220 |709/250)!.CCLS. |(710/303 |710/300 |710/304 |710/302 |710/72 |710/104)!.CCLS. |(235/472.01 |235/472.02)!.CCLS. |(713/300)!.CCLS.)	6415

Database:
US Patents Full-Text Database
US Pre-Grant Publication Full-Text Database
JPO Abstracts Database
EPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search: L1

Refine Search

Recall Text | Clear

Search History

DATE: Wednesday, August 13, 2003 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
	DB=USPT; PLUR=YES; OP=OR		
L1	((708/139)!.CCLS. |(361/683 |361/729 |361/686 |361/727)!.CCLS. |(709/220 |709/250)!.CCLS. |(710/303 |710/300 |710/304 |710/302 |710/72 |710/104)!.CCLS. |(235/472.01 |235/472.02)!.CCLS. |(713/300)!.CCLS.)	6415	L1

END OF SEARCH HISTORY

Your search matched **[0]** of **[961219]** documents.

You may refine your search by editing the current search expression or entering a new one the text box. Then click search Again.

(detect* and dock* and bus)

Search Again

OR

Use your browser's back button to return to your original search page.

Results:

No documents matched your query.

Your search matched **6** of **961219** documents.

A maximum of **6** results are displayed, **25** to a page, sorted by **Relevance** in **descending** order.
You may refine your search by editing the current search expression or entering a new one the text b
Then click **Search Again**.

(detect* and dock*) and (device or module or unit)

Search Again

Results:
Journal or Magazine = **JNL** Conference = **CNF** Standard = **STD**

---

1 **Characterization and application of highly sensitive infra-red emissio microscopy for microprocessor backside failure analysis**
Loh Ter Hoe; Yee Wai Mun; Chew Yin Yan;
Physical and Failure Analysis of Integrated Circuits, 1999. Proceedings of the 1 International Symposium on the , 5-9 July 1999
Page(s): 108 -112

[Abstract] [PDF Full-Text (544 KB)] **IEEE CNF**

---

2 **Failure analysis case studies using the IR-OBIRCH (infrared optical b induced resistance change) method**
Nikawa, K.; Inoue, S.; Morimoto, K.; Sone, S.;
Test Symposium, 1999. (ATS '99) Proceedings. Eighth Asian , 16-18 Nov. 1999
Page(s): 394 -399

[Abstract] [PDF Full-Text (348 KB)] **IEEE CNF**

---

3 **Advanced crane motion control**
Robb, T.;
Industrial Automation and Control: Distributed Control for Automation (Digest 1998/297), IEE Colloquium on , 4 March 1998
Page(s): 6/1 -6/6

[Abstract] [PDF Full-Text (472 KB)] **IEE CNF**

---

4 **Integrated multi-behavior mobile robot navigation using decentralize control**
Tse Min Chen; Luo, R.C.;

Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ Internatio
Conference on , Volume: 1 , 13-17 Oct. 1998
Page(s): 564 -569 vol.1

[Abstract] [PDF Full-Text (560 KB)] **IEEE CNF**

---

5 **Thematic applications of German MOMS-2P images from the MIR spa station**
Bodechtel, J.; Lei, Q.; Frei, M.; Hirscheider, A.;
Geoscience and Remote Sensing Symposium, 1999. IGARSS '99 Proceedings. I 1999 International , Volume: 5 , 28 June-2 July 1999
Page(s): 2584 -2586 vol.5

[Abstract] [PDF Full-Text (468 KB)] **IEEE CNF**

---

6 **Hunting for mines with REMUS: a high performance, affordable, free swimming underwater robot**
von Alt, C.; Allen, B.; Austin, T.; Forrester, N.; Goldsborough, R.; Purcell, M.; S R.;
OCEANS, 2001. MTS/IEEE Conference and Exhibition , Volume: 1 , 5-8 Nov. 20
Page(s): 117 -122 vol.1

[Abstract] [PDF Full-Text (813 KB)] **IEEE CNF**

Integrated multi-behavior mobile robot navigatio using decentralized control

Tse Min Chen Luo, R.C.
Dept. of Electr. Eng., Nat. Chung Cheng Univ.;

This paper appears in: **Intelligent Robots and Systems, 1998. Proceedin 1998 IEEE/RSJ International Conference on**
Meeting Date: 10/13/1998 -10/17/1998
Publication Date: 13-17 Oct 1998
Location: Victoria, BC , Canada
On page(s): 564-569 vol.1
Volume: 1, References Cited: 18
IEEE Catalog Number: 98CH36190
Number of Pages: 3 vol. xliv+2010
INSPEC Accession Number: 6168637

Abstract:
The components for providing autonomous capabilities of a mobile robot are grouped into few basic modules, namely motion planner, motion executor, m assistant, and behavior arbitrator. The primitive motion executors such as ob avoidance, goal following, wall following, docking, and path tracking for mobi robot navigation are developed in this paper. They are integrated with motio planner, motion assistants, and behavior arbitrator together based on decentralized control architecture with a hierarchical shared information mem Event driven concept is used on switching the motion behaviors. The resultan intelligence for mobile robot navigation is capable of efficiently performing m behavior and detecting environmental event in parallel to adapt dynamically changed environment. It also allows the human to program the motion beha in high level to complete a task. Experimental results on a real mobile robot demonstrated the robustness of the motion executors and the overall system

Index Terms:
collision avoidance computerised navigation decentralised control mobile robots rob control tracking behavior arbitrator decentralized control docking environmental ev detection goal following hierarchical shared information memory intelligent mobile rob navigation motion assistant motion assistants motion behavior motion executor mo planner multibehavior mobile robot navigation obstacle avoidance path tracking robustness wall following

WEST

Generate Collection Print

L2: Entry 6 of 22 File: USPT Feb 13, 2001

DOCUMENT-IDENTIFIER: US 6189050 B1
TITLE: Method and apparatus for adding or removing devices from a computer system without restarting

Detailed Description Text (35):
In FIG. 6, when computer 110 is docked with docking station 120, docking port 108-a interfaces with docking port 108-b. The SMI firmware processor 511 then detects the presence, via port link 517-d, of at least the PCI/PCI bridge device 134 that is present on the additional PCI bus 135 in docking station 120. After the PCI/PCI bridge device 134 is detected (step 604), the device type is determined (step 605) and the proper driver (i.e., the PCI/PCI bridge driver 508) is determined to have been preloaded into device driver memory 504. Thread process 502 then creates a symbolic link 560 from the PCI/PCI bridge driver 508 to a location in the file system 501 of the operating system (step 607), and then activates the PCI/PCI bridge driver 508 via signal 561 to either the driver 508 or the operating system 500. Upon the next iteration of the thread process, any devices that are present on the newly added PCI bus 135, such as CD-ROM device 119, are detected and configured in a similar manner by steps 604 through 608.

WEST

Generate Collection Print

L2: Entry 6 of 22 File: USPT Feb 13, 2001

US-PAT-NO: 6189050
DOCUMENT-IDENTIFIER: US 6189050 B1

TITLE: Method and apparatus for adding or removing devices from a computer system without restarting

DATE-ISSUED: February 13, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Sakarda; Premanand	Acton	MA		

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
Compaq Computer Corporation	Houston	TX			02

APPL-NO: 09/ 074767 [PALM]
DATE FILED: May 8, 1998

INT-CL: [07] G06 F 15/40, G06 F 13/22, G06 F 13/00, G06 F 9/445, G06 F 15/02

US-CL-ISSUED: 710/18; 710/103, 710/104, 710/10, 710/15, 710/17, 710/19, 709/301, 709/226, 709/220, 709/300, 713/1, 713/323, 711/115
US-CL-CURRENT: 710/18; 709/220, 709/226, 709/321, 710/10, 710/104, 710/15, 710/17, 710/19, 710/302, 711/115, 713/1, 713/323

FIELD-OF-SEARCH: 710/8, 710/10, 710/18, 710/103, 710/15, 709/100, 709/301, 709/220, 709/226, 395/712, 711/115

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

Search Selected Search ALL

	PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
☐	5319751	June 1994	Garney	711/115
☐	5404494	April 1995	Garney	709/300
☐	5412798	May 1995	Garney	713/1
☐	5630076	May 1997	Saulpaugh et al.	710/10
☐	5655148	August 1997	Richman et al.	710/18
☐	5748980	May 1998	Lipe et al.	710/18
☐	5781798	July 1998	Beatty et al.	710/10
☐	5787246	July 1998	Lichtman et al.	709/220
☐	5793979	August 1998	Lichtman et al.	709/226
☐	5793987	August 1998	Quackenbush et al.	710/100
☐	5802365	September 1998	Kathail et al.	709/301
☐	5809329	September 1998	Lichtman et al.	710/18
☐	5819107	October 1998	Lichtman et al.	710/18
☐	5889965	March 1999	Wallach et al.	710/103
☐	5892928	April 1999	Wallach et al.	710/103
☐	5978923	November 1999	Kou	713/323
☐	6023585	February 2000	Perlman et al.	395/712
☐	6081850	June 2000	Garney	710/15

OTHER PUBLICATIONS

Ioannidis et al. "Porting AIX Onto the Student Electronic Notebook", Proceedings of the 1991 ACM SIGSMALL/PC Symposium on Small Systems, 1991, pp. 76-82.
Banna et al. "Fully Depleted CMOS/SOI Device Design Guidelines for Low-Power Applications", Electron Devices, IEEE Transactions on, vo. 46 Issue: Apr. 4, 1999, pp. 754-761.

ART-UNIT: 272

PRIMARY-EXAMINER: Lee; Thomas C.

ASSISTANT-EXAMINER: Schuster; Katharina

ATTY-AGENT-FIRM: Hamilton, Brook, Smith & Reynolds, P.C.

ABSTRACT:

A method and apparatus provide a mechanism for a personal computer to allow the insertion and removal of devices to and from device ports without re-starting the operating system of the computer. Device drivers are pre-loaded during the start-up process of the computer system for devices that may be inserted in the system later. Upon detection of device insertion, a high priority thread process determines the type of device inserted and determines which pre-loaded device driver can operate the newly inserted device. The selected device driver is linked to the file system and is activated by signaling to the device driver or to the operating system of the existence of the newly inserted device. The operating system can then operate the inserted device. Upon device removal, device drivers for the removed device can be deactivated. The system also allows a docking station to have devices added or removed from device ports after the docking process has been completed.

20 Claims, 6 Drawing figures

WEST

Generate Collection | Print

L2: Entry 14 of 22 File: USPT Feb 2, 1999

DOCUMENT-IDENTIFIER: US 5867406 A
TITLE: Docking device for a portable computer and a method for docking a portable computer to the docking device

Detailed Description Text (31):
With reference now to FIG. 7, the process begins at block 100 when the connectors for connecting host 50 and docking device 60 are mechanically coupled by the user. Since the interface circuit 41 and the CPU 42 communicating with the subbus 34 within docking device 60 are already in an operating state, the mechanical connection can be detected by a change in DOCKED# to ACTIVE.sub.-- LOW as illustrated at step 102. Then, at block 104 it is determined from NOTE.sub.-- ID0 whether host 50 is a new type or an old type of portable computer. If host 50 is an old type, the SMI from the docking device 60 cannot be processed as described above, and thus system bus 4 cannot be electrically connected in the same operation. Accordingly, the process branches to off-page connector S.

WEST

Generate Collection | Print

L2: Entry 14 of 22 File: USPT Feb 2, 1999

US-PAT-NO: 5867406
DOCUMENT-IDENTIFIER: US 5867406 A

TITLE: Docking device for a portable computer and a method for docking a portable computer to the docking device

DATE-ISSUED: February 2, 1999

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Yanagisawa; Takashi	Yokohama			JP

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
International Business Machines Corp.	Armonk	NY			02

APPL-NO: 08/ 416398 [PALM]
DATE FILED: April 6, 1995

FOREIGN-APPL-PRIORITY-DATA:

COUNTRY	APPL-NO	APPL-DATE
JP	6-134124	June 16, 1994

INT-CL: [06] G06 F 7/50

US-CL-ISSUED: 364/708.1; 361/683, 361/729, 361/733
US-CL-CURRENT: 708/140; 361/683, 361/729, 361/733

FIELD-OF-SEARCH: 364/708.1, 361/680-686, 361/728-733, 395/281, 395/283, 395/500, 395/800

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

Search Selected | Search ALL

	PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
☐	5030128	July 1991	Herron et al.	439/372
☐	5123092	June 1992	Buxton et al.	395/250
☐	5265238	November 1993	Canova, Jr. et al.	395/500
☐	5323291	June 1994	Boyle et al.	361/683
☐	5377357	December 1994	Nishigaki et al.	395/800
☐	5430883	July 1995	Horiuchi	395/750
☐	5463742	October 1995	Kobayashi	395/281
☐	5473499	December 1995	Weir	361/58
☐	5526493	June 1996	Shu	395/281

ART-UNIT: 276

PRIMARY-EXAMINER: Moise; Emmanuel L.

ATTY-AGENT-FIRM: Dillon; Andrew J.

ABSTRACT:

A docking device for a portable computer includes a body adapted to receive a portable computer. The body encloses a number of electronic components adapted to be connected to a portable computer via a number of signal lines. A connector is provided in the body which mechanically connects the number of signal lines within the body to a corresponding number of signal lines within the portable computer. The docking device also includes a signal connection means interposed between the portable computer and the electronic components for selectively electrically interconnecting the particular ones of the signal lines within the body to corresponding signal lines among the number of signal lines within the portable computer in response to an operating state of said portable computer. A preferred embodiment of the docking device further includes a control means for controlling the selective electrical interconnection of the particular ones of the number of signal lines by the signal connection means. In this preferred embodiment, the control means electrically interconnects the particular ones of the number of signal lines within the body to corresponding signal lines within the portable computer in response to an acknowledgment by the portable computer of a request by the control means to electrically interconnect the particular ones of the number of signal lines.

17 Claims, 23 Drawing figures

WEST

Generate Collection Print

L6: Entry 29 of 33 File: USPT Nov 26, 1996

DOCUMENT-IDENTIFIER: US 5579487 A
TITLE: Portable work slate computer with multiple docking positions for interchangeably receiving removable modules

Abstract Text (1):
A configurable electronic work slate unit includes a customizable array of data devices and input/output devices which are selectively integrated in a compact and highly ergonomic structure. Increased operator productivity is accomplished by use of several possible user interface media including a multi-function display and input/output unit including a digitizer, position sensitive screen, and video display panel. Selectively integratable user interface components include a bar code scanner, RF modulator for radio frequency communication, modem, audio input/output, as well as the multi-purpose display. All components are integrated by use of a handle unit or module docking assembly which can house a battery as well as removable modules in a manner so as to minimize operator fatigue and discomfort, as well as enhance productivity. Configuration, reconfiguration and updating of work slate units by choice and selection of removable modules, from a variety of different types of modules, is enhanced by utilization of standardized interfaces which are intercoupled to a common signal bus. The bus configuration in combination with a bus controller functions as an intelligent bus capable of identifying types of removable modules and configuring communication protocols for operation with different types of removable modules.

Detailed Description Text (56):
In accordance with the invention, each removable module, in addition to providing one or more operating capabilities, also includes some provision for enabling its type to be identified. As an example, a certain interface path can be utilized to access a coded or other identifier within the removable module. As a result, when a removable module is first placed in a docking position bus controller 202 is enabled to access the coded identifier in order to identify the type of removable module. Having identified the type of module, the bus controller 202 is arranged to provide information based upon such identification to the CPU and possibly other elements of the electronic unit to enable communication with the module by use of the appropriate protocol. Thus, while the interfaces are physically standardized, the interface and bus usage are configured as appropriate for the particular removable module involved. With an understanding of the invention, it will be apparent that the bus controller may take the form of a separate included element as shown in FIG. 11, may be included as a part of a CPU processor unit, may take the form of a removable module, etc. Given the current state of local network bus control and configuration technology and computation and signal recognition technology in general, skilled persons once having an understanding of the invention will be able to provide a variety of module identification, protocol activation and bus configuration approaches as appropriate for a variety of applications.

WEST

Generate Collection | Print

L6: Entry 29 of 33 File: USPT Nov 26, 1996

US-PAT-NO: 5579487
DOCUMENT-IDENTIFIER: US 5579487 A

TITLE: Portable work slate computer with multiple docking positions for interchangeably receiving removable modules

DATE-ISSUED: November 26, 1996

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Meyerson; Robert F.	Captiva Island	FL		
Chang; Yung-Fu	Medina	OH		
Wang; Ynjiun P.	Ft. Myers	FL		
Wall; Daniel G.	Union Town	OH		

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
Teletransaction, Inc.	Akron	OH			02

APPL-NO: 08/ 269190 [PALM]
DATE FILED: June 30, 1994

PARENT-CASE:
This is a continuation in part of application Ser. No. 07/956,112, filed Oct. 2, 1992, now abandoned.

INT-CL: [06] G06 F 13/00

US-CL-ISSUED: 395/280; 395/282, 395/651, 395/800
US-CL-CURRENT: 710/100; 361/686, 710/303, 713/1

FIELD-OF-SEARCH: 395/282, 395/280, 395/700, 395/800

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

Search Selected | Search ALL

PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
4683550	July 1987	Jindrick et al.	364/900
5377357	December 1994	Nishigaki et al.	395/800
5386567	January 1995	Lien et al.	395/700
5392447	February 1995	Schlack et al.	395/800

ART-UNIT: 235

PRIMARY-EXAMINER: Harvey; Jack B.

ASSISTANT-EXAMINER: Travis; John

ATTY-AGENT-FIRM: Watts, Hoffmann, Fisher & Heinke Co

ABSTRACT:

A configurable electronic work slate unit includes a customizable array of data devices and input/output devices which are selectively integrated in a compact and highly ergonomic structure. Increased operator productivity is accomplished by use of several possible user interface media including a multi-function display and input/output unit including a digitizer, position sensitive screen, and video display panel. Selectively integratable user interface components include a bar code scanner, RF modulator for radio frequency communication, modem, audio input/output, as well as the multi-purpose display. All components are integrated by use of a handle unit or module docking assembly which can house a battery as well as removable modules in a manner so as to minimize operator fatigue and discomfort, as well as enhance productivity. Configuration, reconfiguration and updating of work slate units by choice and selection of removable modules, from a variety of different types of modules, is enhanced by utilization of standardized interfaces which are intercoupled to a common signal bus. The bus configuration in combination with a bus controller functions as an intelligent bus capable of identifying types of removable modules and configuring communication protocols for operation with different types of removable modules.

12 Claims, 23 Drawing figures

WEST

Generate Collection | Print

Search Results - Record(s) 1 through 10 of 19 returned.

☐ 1. Document ID: US 6567876 B1

L3: Entry 1 of 19 File: USPT May 20, 2003

US-PAT-NO: 6567876
DOCUMENT-IDENTIFIER: US 6567876 B1

TITLE: Docking PCI to PCI bridge using IEEE 1394 link

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw Desc | Image

---

☐ 2. Document ID: US 6519669 B1

L3: Entry 2 of 19 File: USPT Feb 11, 2003

US-PAT-NO: 6519669
DOCUMENT-IDENTIFIER: US 6519669 B1

TITLE: Apparatus and method of connecting a computer and a peripheral device

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw Desc | Image

---

☐ 3. Document ID: US 6460106 B1

L3: Entry 3 of 19 File: USPT Oct 1, 2002

US-PAT-NO: 6460106
DOCUMENT-IDENTIFIER: US 6460106 B1

TITLE: Bus bridge for hot docking in a portable computer system

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw Desc | Image

---

☐ 4. Document ID: US 6356963 B1

L3: Entry 4 of 19 File: USPT Mar 12, 2002

US-PAT-NO: 6356963
DOCUMENT-IDENTIFIER: US 6356963 B1

TITLE: Long latency interrupt handling and input/output write posting

WEST

Generate Collection | Print

Search Results - Record(s) 11 through 19 of 19 returned.

☐ 11. Document ID: US 5933609 A

L3: Entry 11 of 19 File: USPT Aug 3, 1999

US-PAT-NO: 5933609
DOCUMENT-IDENTIFIER: US 5933609 A

TITLE: Method and system for hot docking a portable computer to a docking station via the primary PCI bus

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC
Draw Desc | Image

☐ 12. Document ID: US 5920728 A

L3: Entry 12 of 19 File: USPT Jul 6, 1999

US-PAT-NO: 5920728
DOCUMENT-IDENTIFIER: US 5920728 A

TITLE: Dynamic hibernation time in a computer system

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC
Draw Desc | Image

☐ 13. Document ID: US 5889964 A

L3: Entry 13 of 19 File: USPT Mar 30, 1999

US-PAT-NO: 5889964
DOCUMENT-IDENTIFIER: US 5889964 A

TITLE: Method and apparatus for docking and undocking a notebook computer to and from a docking station while the notebook computer is in an active state

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC
Draw Desc | Image

☐ 14. Document ID: US 5884049 A

L3: Entry 14 of 19 File: USPT Mar 16, 1999

US-PAT-NO: 5884049
DOCUMENT-IDENTIFIER: US 5884049 A

TITLE: Increased processor performance comparable to a desktop computer from a docked portable computer

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC
Draw Desc | Image

☐ 15. Document ID: US 5875307 A

L3: Entry 15 of 19 File: USPT Feb 23, 1999

US-PAT-NO: 5875307
DOCUMENT-IDENTIFIER: US 5875307 A

TITLE: Method and apparatus to enable docking/undocking of a powered-on bus to a docking station

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC
Draw Desc | Image

☐ 16. Document ID: US 5867406 A

L3: Entry 16 of 19 File: USPT Feb 2, 1999

US-PAT-NO: 5867406
DOCUMENT-IDENTIFIER: US 5867406 A

TITLE: Docking device for a portable computer and a method for docking a portable computer to the docking device

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC
Draw Desc | Image

☐ 17. Document ID: US 5862349 A

L3: Entry 17 of 19 File: USPT Jan 19, 1999

US-PAT-NO: 5862349
DOCUMENT-IDENTIFIER: US 5862349 A

TITLE: Method and apparatus for docking and undocking a notebook computer

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC
Draw Desc | Image

☐ 18. Document ID: US 5761460 A

L3: Entry 18 of 19 File: USPT Jun 2, 1998

US-PAT-NO: 5761460
DOCUMENT-IDENTIFIER: US 5761460 A

TITLE: Reconfigurable dual master IDE interface

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC
Draw Desc | Image

☐ 19. Document ID: US 5323291 A

L3: Entry 19 of 19 File: USPT Jun 21, 1994

US-PAT-NO: 5323291
DOCUMENT-IDENTIFIER: US 5323291 A

TITLE: Portable computer and docking station having an electromechanical docking/undocking mechanism and a plurality of cooperatively interacting failsafe mechanisms

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC
Draw Desc | Image

Generate Collection | Print

Terms	Documents
l1 and L2	19

Display Format: TI | Change Format

Previous Page Next Page

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC
Draw Desc | Image

☐ 5. Document ID: US 6189050 B1

L3: Entry 5 of 19 File: USPT Feb 13, 2001

US-PAT-NO: 6189050
DOCUMENT-IDENTIFIER: US 6189050 B1

TITLE: Method and apparatus for adding or removing devices from a computer system without restarting

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC
Draw Desc | Image

☐ 6. Document ID: US 6062480 A

L3: Entry 6 of 19 File: USPT May 16, 2000

US-PAT-NO: 6062480
DOCUMENT-IDENTIFIER: US 6062480 A

TITLE: Hot docking system and methods for detecting and managing hot docking of bus cards

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC
Draw Desc | Image

☐ 7. Document ID: US 6038624 A

L3: Entry 7 of 19 File: USPT Mar 14, 2000

US-PAT-NO: 6038624
DOCUMENT-IDENTIFIER: US 6038624 A

TITLE: Real-time hardware master/slave re-initialization

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC
Draw Desc | Image

☐ 8. Document ID: US 5991839 A

L3: Entry 8 of 19 File: USPT Nov 23, 1999

US-PAT-NO: 5991839
DOCUMENT-IDENTIFIER: US 5991839 A

TITLE: Computer system having computer main body and expansion unit

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC
Draw Desc | Image

☐ 9. Document ID: US RE36381 E

L3: Entry 9 of 19 File: USPT Nov 9, 1999

US-PAT-NO: RE36381
DOCUMENT-IDENTIFIER: US RE36381 E

TITLE: Portable computer and docking station having an electromechanical docking/undocking mechanism and a plurality of cooperatively interacting failsafe mechanisms

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC
Draw Desc | Image

☐ 10. Document ID: US 5935226 A

L3: Entry 10 of 19 File: USPT Aug 10, 1999

US-PAT-NO: 5935226
DOCUMENT-IDENTIFIER: US 5935226 A

TITLE: Method and apparatus for issuing transaction requests to a target device in accordance with the state of connection between the portable computer and the target device

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC
Draw Desc | Image

Generate Collection | Print

Terms	Documents
l1 and L2	19

Display Format: TI | Change Format

Previous Page Next Page

WEST

Generate Collection | Print

L6: Entry 1 of 33 File: USPT Apr 15, 2003

DOCUMENT-IDENTIFIER: US 6549416 B2
TITLE: Portable computer docking station with protected connector

Abstract Text (1):
The docking station includes one or more ramps which receive the computer during docking. The ramps are elevated so as to shield a docking connector of the docking station during reception of the computer. With the computer positioned on the ramps, the ramps are lowered to bring the computer into engagement with the station's docking connector. The computer is secured in the docked position by any of a coupling force between docking connectors, a latching action applied to the ramps, or a latching action applied to the computer.

Detailed Description Text (2):
Referring to FIG. 1, a docking environment includes a portable computer 10, a docking station 12 and one or more peripheral devices 14. The docking station 12 provides a convenient interface for connecting the portable computer 10 to one or more desktop peripherals and/or a network interface. The portable computer 10 is a general purpose, programmed portable computing system of the type which is well known in the art. For example the computer 10 is of the notebook computer type in one embodiment and of the subnotebook computer type in another embodiment. The portable computer system 10 has a display panel 15, a keyboard 16, a pointing device 17, a clicking device 18, a system board 20 with a central processing unit (CPU) 22 and random access memory (RAM) 24 and a docking interface 34 terminating at a docking connector 36. Typically the computer 10 also includes a hard disk drive 26 with hard disk, and optionally--one or more network interfaces 28 (e.g., modem, ethernet adapter, infrared adapter), and one or more transportable storage media drives 30 and media (e.g., CD-ROM drive, DVD-ROM drive, floppy disk drive, zip drive, bernoulli drive). For example, the computer 10 in some embodiments includes an expansion bay into which a removable media drive is installed. The various components interface and exchange data and commands through one or more busses 32. The computer system 10 receives information by entry through the keyboard 14, pointing/clicking devices 16/18, the network interface 28 or another input device or input port.

WEST

Generate Collection | Print

L6: Entry 1 of 33 File: USPT Apr 15, 2003

US-PAT-NO: 6549416
DOCUMENT-IDENTIFIER: US 6549416 B2

TITLE: Portable computer docking station with protected connector

DATE-ISSUED: April 15, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Sterner; John R.	Albany	OR		
Singleton, Jr.; Charles W.	Corvallis	WA		

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
Hewlett Packard Development Company, L.P.	Houston	TX			02

APPL-NO: 09/ 809381 [PALM]
DATE FILED: March 15, 2001

INT-CL: [07] H05 K 7/12, G06 F 1/16

US-CL-ISSUED: 361/727; 361/683, 361/726, 361/801, 439/341
US-CL-CURRENT: 361/727; 361/683, 361/726, 361/801, 439/341

FIELD-OF-SEARCH: 361/679, 361/683, 361/724-727, 361/801, 361/802, 439/341, 439/372, 439/376, 439/928.1

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

Search Selected | Search ALL

PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
4969830	November 1990	Daly et al.	200/50.1
5402310	March 1995	Penniman	361/686
5805412	September 1998	Yanagisawa et al.	361/686
5870283	February 1999	Maeda et al.	361/686
6072695	June 2000	Steiger et al.	361/686
6093039	July 2000	Lord	361/686
6135801	October 2000	Helot et al.	439/341
6264488	July 2001	Helot et al.	361/686

ART-UNIT: 2841

PRIMARY-EXAMINER: Gandhi; Jayprakash N.

ABSTRACT:

The docking station includes one or more ramps which receive the computer during docking. The ramps are elevated so as to shield a docking connector of the docking station during reception of the computer. With the computer positioned on the ramps, the ramps are lowered to bring the computer into engagement with the station's docking connector. The computer is secured in the docked position by any of a coupling force between docking connectors, a latching action applied to the ramps, or a latching action applied to the computer.

27 Claims, 12 Drawing figures

WEST

Generate Collection Print

L6: Entry 4 of 33 File: USPT Nov 21, 2000

DOCUMENT-IDENTIFIER: US 6151646 A
TITLE: System for resources under control of docking station when stand alone and resources under control of central processing unit of portable computer when docked

Abstract Text (1):
The described embodiments of the present invention provide a computer docking station (12, 32, 58, 68, 76, 84, 90, 92, 94, 96) that can have its functionality reconfigured by a docked portable personal computer (10, 38, 62, 66, 74, 82). In at least one embodiment of the invention, the computer docking station is configured as a stand alone computer prior to docking with a portable computer, may have its functionality reconfigured when docked to the portable computer, and reconfigures itself to be a stand alone computer when undocked from the portable computer. In one embodiment of the invention, docking station resources are placed under the control of a docked portable computer. In another embodiment of the invention, docked portable computer resources are placed under the control of the docking station. The invention contemplates docking via direct connection, radio requency "RF" communications, infrared "IR" communications, 1394 high performance serial bus communications, or card bus communications, and/or combinations of one or more of these communications techniques.

Detailed Description Text (47):
MUX 24 is fairly straight forward. It is really a switch with date lines that signal whether or not there is a docked notebook computer to take control over some of the docking station resources. One method pulls a pin high to provide this function. When notebook 10 hard docks to docking station 12, it will pull the in high on the mux which indicates a docked situation. System card 28 on the other side of mux 24 reads the I/O device and signals that it is being docked. The system pulls mux 24 and lets the cpu in notebook 10 have control over the released resources in docking station 12. One way of accomplishing this is to write simple code to the cpu in system card 28 that says "if there is an input (interrupt) from the mux (which you could hook to external interrupt 1 or 2 off of the SMI interrupt on the CPU), the codes see this and says, "read the status--docked or undocked". If docked, you turn over the devices, the CPU quits running on the PCI bus (it doesn't issue any PCI cycles) and throws the mux so that the PCI bus is being driven by the external master and not the CPU in the card. When the mux goes back, then the CPU starts driving the internal PCI bus. Thus, in a docked situation with a notebook, notebook may be given partial or full control over the resources of system card 28.

Detailed Description Text (218):
In yet another embodiment of the invention, illustrated in FIG. 264, at least one additional card slot (three actually shown) ise added to the docking station of FIG. 3. Each additional card slot 26 is coupled to mux 24. Each additional card slot is coupleable to a card 30. Each card 30 contains at least one resource that is an addition to the resources available on system card 28. Being that cards 30 couple to mux 24, the cpu in the docking station (on system card 28 in this embodiment), controls the resources of cards 30 when the docking station is "undocked". When notebook 10 docks to docking station 12, as illustrated in FIG. 265, it will pull the pin high on the mux which indicated a docked situation. System card 28 on the other side of mux 24 reads the I/O device and signals that it is being docked. The system pulls mux 24 and lets the cpu in notebook 10 have control over the released resources in docking station 12, be they resources on system card 28 and/or cards(s) 30. As discussed previously, one way of accomplishing this is to write simple code

to the cpu in system card 28 that says "if there is an input (interrupt) from the mux (which you could hook to external interrupt 1 or 2 off of the SMI interrupt on the CPU), the codes see this and says, "read the status--docked or undocked". If docked, the docking station turns over the resources, the CPU quits running on the PCI bus 14 (it doesn't issue any PCI cycles) and throws the mux so that the PCI bus is being driven by the external master and not the CPU in the system card 28. When the mux goes back, then the CPU in system card 28 starts driving the internal PCI bus 14. Thus, in a docked situation with a notebook, notebook may be given partial or full control over the resources of system card 28 and additional cards 30.

Detailed Description Text (285):
In yet another embodiment or the invention, illustrated in FIG. 327, at least one additional card slot (three actually shown) are added to the docking station of FIG. 319. Each card slot 26 is coupled to mux 24. Each additional card slot is coupleable to a card 30. Each card 30 is treated as a system resource. When notebook 38 docks to docking station 32, as illustrated in FIG. 328, it will pull the pin high on the mux which indicates a docked situation. System card 28 on the other side of mux 24 reads the I/O device and signals that it is being docked. The system pulls mux 24 and lets the cpu in notebook 10 have control over the released resources in docking station 32. As discussed previously, one way of accomplishing this is to write simple code to the cpu in system card 28 that says "if there is an input (interrupt) from the mux (which you could hook to external interrupt 1 or 2 off of the SMI interrupt on the CPU), the codes see this and says, "read the status--docked or undocked". If docked, the docking station turns over the resources, the CPU quits running on the PCI bus 14 (it doesn't issue any PCI cycles) and throws the mux so that the PCI bus is being driven by the external master and not the CPU in the system card 28. When the mux goes back, then the CPU in system card 28 starts driving the internal PCI bus 14. Thus, in a docked situation with a notebook, notebook may be given partial or full control over the resources of system card 28 and additional cards 30.

Detailed Description Text (287):
In yet another embodiment of the invention, illustrated in FIG. 329, at least one additional card slot (three actually shown) are added to the docking station of FIG. 325. Each card slot 26 is coupled to mux 24. Each additional card slot is coupleable to a card 30. Each card 30 is treated as a system resource. When notebook 38 docks to docking station 32, as illustrated in FIG. 330, it will pull the pin high on the mux which indicates a docked situation. System card 28 on the other side of mux 24 reads the I/O device and signals that it is being docked. The system pulls mux 24 and lets the cpu in notebook 10 have control over the released resources in docking station 12. As discussed previously, one way of accomplishing this is to write simple code to the cpu in system card 28 that says "if there is an input (interrupt) from the mux (which you could hook to external interrupt 1 or 2 off of the SMI interrupt on the CPU), the codes see this and says, "read the status--docked or undocked". If docked, the docking station turns over the resources, the CPU quits running on the PCI bus 14 (it doesn't issue any PCI cycles) and throws the mux so that the PCI bus is being driven by the external master and not the CPU in the system card 28. When the mux goes back, then the CPU in system card 28 starts driving the internal PCI bus 14. Thus, in a docked situation with a notebook, notebook may be given partial or full control over the resources of system card 28 and additional cards 30.

Detailed Description Text (307):
In yet another embodiment of the invention, illustrated in FIG. 337, at least one additional card slot (three actually shown) are added to the docking station of FIG. 331. Each card slot 26 is coupled to mux 24. Each additional card slot is coupleable to a card 30. Each card 30 is treated as a system resource. When notebook 62 docks to docking station 58, as illustrated in FIG. 338, it will pull the pin high on the mux which indicates a docked situation. System card 28 on the other side of mux 24 reads the I/O device and signals that it is being docked. The system pulls mux 24 and lets the cpu in notebook 10 have control over the released resources in docking station 58. As discussed previously, one way of accomplishing this is to write simple code to the cpu in system card 28 that says "if there is an input (interrupt) from the mux (which you could hook to external interrupt 1 or 2 off of the SMI interrupt on the CPU), the codes see this and says, "read the status--docked or

undocked". If docked, the docking station turns over the resources, the CPU quits running on the PCI bus 14 (it doesn't issue any PCI cycles) and throws the mux so that the PCI bus is being driven by the external master and not the CPU in the system card 28. When the mux goes back, then the CPU in system card 28 starts driving the internal PCI bus 14. Thus, in a docked situation with a notebook, notebook may be given partial or full control over the resources of system card 28 and additional cards 30.

Detailed Description Text (309):
In yet another embodiment of the invention, illustrated in FIG. 339, at least one additional card slot (three actually shown) are added to the docking station of FIG. 335. Each card slot 26 is coupled to mux 24. Each additional card slot is coupleable to a card 30. Each card 30 is treated as a system resource. When notebook 38 docks to docking station 58, as illustrated in FIG. 336, it will pull the pin high on the mux which indicates a docked situation. System card 28 on the other side of mux 24 reads the I/O device and signals that it is being docked. The system pulls mux 24 and lets the cpu in notebook 10 have control over the released resources in docking station 58. As discussed previously, one way of accomplishing this is to write simple code to the cpu in system card 28 that says "if there is an input (interrupt) from the mux (which you could hook to external interrupt 1 or 2 off of the SMI interrupt on the CPU), the codes see this and says, "read the status--docked or undocked". If docked, the docking station turns over the resources, the CPU quits running on the PCI bus 14 (it doesn't issue any PCI cycles) and throws the mux so that the PCI bus is being driven by the external master and not the CPU in the system card 28. When the mux goes back, then the CPU in system card 28 starts driving the internal PCI bus 14. Thus, in a docked situation with a notebook, notebook may be given partial or full control over the resources of system card 28 and additional cards 30.

Detailed Description Text (321):
In yet another embodiment of the invention, illustrated in FIG. 346, at least one additional card slot (three actually shown) are added to the docking station of FIG. 341. Each card slot 26 is coupled to mux 24. Each additional card slot is coupleable to a card 30. Each card 30 is treated as a system resource. When notebook 66 docks to docking station 68, as illustrated in FIG. 347, it will pull the pin high on the mux which indicates a docked situation. System card 28 on the other side of mux 24 reads the I/O device and signals that it is being docked. The system pulls mux 24 and lets the cpu in notebook 66 have control over the released resources in docking station 68. As discussed previously, one way of accomplishing this is to write simple code to the cpu in system card 28 that says "if there is an input (interrupt) from the mux (which you could hook to external interrupt 1 or 2 off of the SMI interrupt on the CPU), the codes see this and says, "read the status--docked or undocked". If docked, the docking station turns over the resources, the CPU quits running on the PCI bus 14 (it doen't issue any PCI cycles) and throws the mux so that the PCI bus is being driven by the external master and not the CPU in the system card 28. When the mux goes back, then the CPU in system card 28 starts driving the internal PCI bus 14. Thus, in a docked situation with a notebook, notebook may be given partial or full control over the resources of system card 28 and additional cards 30.

Detailed Description Text (323):
In yet another embodiment of the invention, illustrated in FIG. 348, at least one additional card slot (three actually shown) are added to the docking station of FIG. 344. Each card slot 26 is coupled to mux 24. Each additional card slot is coupleable to a card 30. Each card 30 is treated as a system resource. When notebook 66 docks to docking station 68, illustrated in FIG. 349, it will pull the pin high on the mux which indicates a docked situation. System card 28 on the other side of mux 24 reads the I/O device and signals that it is being docked. The system pulls mux 24 and lets the cpu in notebook 66 have control over the released resources in docking station 68. As discussed previously, one way of accomplishing this is to write simple code to the cpu in system card 28 that says "if there is an input (interrupt) from the mux (which you could hook to external interrupt 1 or 2 off of the SMI interrupt on the CPU), the codes see this and says, "read the status--docked or undocked". If docked, the docking station turns over the resources, the CPU quits running on the PCI bus 14 (it doesn't issue any PCI cycles) and throws the mux so that the PCI bus

is being driven by the external master and not the CPU in the system card 28. When the mux goes back, then the CPU in system card 28 starts driving the internal PCI bus 14. Thus, in a docked situation with a notebook, notebook may be given partial or full control over the resources of system card 28 and additional cards 30.

Detailed Description Text (331):
In yet another embodiment of the invention, illustrated in FIG. 355, at least one additional card slot (three actually shown) are added to the docking station of FIG. 350. Each card slot 26 is coupled to mux 24. Each additional card slot is coupleable to a card 30. Each card 30 is treated as a system resource. When notebook 74 docks to docking station 76, as illustrated in FIG. 356, it will pull the pin high on the mux which indicates a docked situation. System card 28 on the other side of mux 24 reads the I/O device and signals that it is being docked. The system pulls mux 24 and lets the cpu in notebook 74 have control over the released resources in docking station 76. As discussed previously, one way of accomplishing this it to write simple code to the cpu in system card 28 that says "if there is an input (interrupt) from the mux (which you could hook to external interrupt 1 or 2 off of the SMI interrupt on the CPU), the codes see this and says, "read the status--docked or undocked". If docked, the docking station turns over the resources, the CPU quits running on the PCI bus 14 (it doesn't issue any PCI cycles) and throws the mux so that the PCI bus is being driven by the external master and not the CPU in the system card 28. When the mux goes back, then the CPU in system card 28 starts driving the internal PCI bus 14. Thus, in a docked situation with a notebook, notebook may be given partial or full control over the resources of system card 28 and additional cards 30.

Detailed Description Text (333):
In yet another embodiment of the invention, illustrated in FIG. 357, at least one additional card slot (three actually shown) are added to the docking station of FIG. 353. Each card slot 26 is coupled to mux 24. Each additional card slot is coupleable to a card 30. Each card 30 is treated as a system resource. When notebook 74 docks to docking station 75, illustrated in FIG. 353, it will pull the pin high on the mux which indicates a docked situation. System card 28 on the other side of mux 24 reads the I/O device and signals that it is being docked. The system pulls mux 24 and lets the cpu in notebook 74 have control over the released resources in docking station 76. As discussed previously, one way of accomplishing this is to write simple code to the cpu in system card 28 that says "if there is an input (interrupt) from the mux (which you could hook to external interrupt 1 or 2 off of the SMI interrupt on the CPU), the codes see this and says, "read the status--docked or undocked". If docked, the docking station turns over the resources, the CPU quits running on the PCI bus 14 (it doesn't issue any PCI cycles) and throws the mux so that the PCI bus is being driven by the external master and not the CPU in the system card 28. When the mux goes back, then the CPU in system card 28 starts driving the internal PCI bus 14. Thus, in a docked situation with a notebook, notebook may be given partial or full control over the resources of system card 28 and additional cards 30.

Detailed Description Text (342):
In yet another embodiment of the invention, illustrated in FIG. 364, at least one additional card slot (three actually shown) are added to the docking station of FIG. 359. Each card slot 26 is coupled to mux 24. Each additional card slot is coupleable to a card 30. Each card 30 is treated as a system resource. When notebook 82 docks to docking station 84, as illustrated in FIG. 365, it will pull the pin high on the mux which indicates a docked situation. System card 28 on the other side of mux 24 reads the I/O device and signals that it is being docked. The system pulls mux 24 and lets the cpu in notebook 10 have control over the released resources in docking station 84. As discussed previously, one way of accomplishing this is to write simple code to the cpu in system card 28 that says "if there is an input (interrupt) from the mux (which you could hook to external interrupt 1 or 2 off of the SMI interrupt on the CPU), the codes see this and says, "read the status--docked or undocked". If docked, the docking station turns over the resources, the CPU quits running on the PCI bus 14 (it doesn't issue any PCI cycles) and throws the mux so that the PCI bus is being driven by the external master and not the CPU in the system card 28. When the mux goes back, then the CPU in system card 28 starts driving the internal PCI bus 14. Thus, in a docked situation with a notebook may be given partial or full control over the resources of system card 28 and additional

cards 30.

Detailed Description Text (344):
In yet another embodiment of the invention, illustrated in FIG. 366, at least one additional card slot (three actually shown) are added to the docking station of FIG. 362. Each card slot 26 is coupled to mux 24. Each additional card slot is coupleable to a card 30. Each card 30 is treated as a system resource. When notebook 82 docks to docking station 84, illustrated in FIG. 367, it will pull the pin high on the mux which indicates as docked situation. System card 28 on the other side of mux 24 reads the I/O device and signals that it is being docked. The system pulls mux 24 and lets the cpu in notebook 10 have control over the released resources in docking station 58. As discussed previously, one way of accomplishing this to write simple code to the cpu in system card 28 that says "if there is an input (interrupt) from the mux (which you could hook to external interrupt 1 or 2 off of the SMI interrupt on the CPU), the codes see this and says, "read the status--docked or undocked". If docked, the docking station turns over the resources, the CPU quits running on the PCI bus 14 (it doesn't issue any PCI cycles) and throws the mux so that the PCI bus is being driven by the external master and not the CPU in the system card 28. When the mux goes back, then the CPU in system card 28 starts driving the internal PCI bus 14. Thus, in a docked situation with a notebook, notebook may be given partial or full control over the resources of system 28 and additional cards 30.

Detailed Description Text (356):
In yet other embodiments of the invention, illustrated in FIGS. 370, 374, 378 and 382, at least one additional card slot (three actually shown) are added to the docking stations of FIGS. 368, 372, 376 and 380, respectively. Each card slot 26 is coupled to mux 24. Each additional card slot is coupleable to a card 30. Each card 30 is treated as a system resource. When a notebook docks to docking station 90 it will pull the pin high on the mux which indicates a docked situation. System card 28 on the other side of mux 24 reads the I/O device and signals that it is being docked. The system pulls mux 24 and lets the cpu in notebook have control over the released resources in docking station 90. As discussed previously, one way of accomplishing this is to write simple code to the cpu in system card 28 that says "if there is an input (interrupt) from the mux (which you could hook to external interrupt 1 or 2 off of the SMI interrupt on the CPU), the codes see this and says, "read the status--docked or undocked". If docket, the docking station turns over the resources, the CPU quits running on the PCI bus 14 (it doesn't issue any PCI cycles) and throws the mux so that the PCI bus is being driven by the external master and not the CPU in the system card 28. When the mux goes back, then the CPU in system card 28 starts driving the internal PCI bus 14. Thus, in a docked situation with a notebook, notebook may be given partial or full control over the resources of system card 28 and additional cards 30.

Detailed Description Text (363):
In yet other embodiments of the invention, illustrated in FIGS. 386, 390, 394 and 398, at least one additional card slot (three actually shown) are added to the docking stations of FIGS. 384, 388, 392 and 396. respectively. Each card slot 26 is coupled to mux 24. Each additional card slot is coupleable to a card 30. Each card 30 is treated as a system resource. When a notebook docks to docking station 92 it will pull the pin high on the mux which indicates a docked situation. System card 28 on the other side of mux 24 reads the I/O device and signals that it is being docked. The system pulls mux 24 and lets the cpu in notebook have control over the released resources in docking station 92. As discussed previously, one way of accomplishing this is to write simple code to the cpu in system card 28 that says "if there is an input (interrupt) from the mux (which you could hook to external interrupt 1 or 2 off of the SMI interrupt on the CPU), the codes see this and says, "read the status--docked or undocked". If docked, the docking station turns over the resources, the CPU quits running on the PCI bus 14 (it doesn't issue any PCI cycles) and throws the mux so that the PCI bus is being driven by the external master and not the CPU in the system card 28. When the mux goes back, then the CPU in system card 28 starts driving the internal PCI bus 14. Thus, in a docked situation with a notebook, notebook may be given partial or full control over the resources of system card 28 and additional cards 30.

Detailed Description Text (365):

In still other embodiments of the invention, illustrated in FIGS. 387, 391, 395, and 399, at least one additional card slot (three actually shown) are added to the docking station of FIG. 385. Each card slot 26 is coupled to mux 24. Each additional card slot is coupleable to a card 30. Each card 30 is treated as a system resource. When a notebook having at least one corresponding interface docks to docking station 92, it will pull the pin high on the mux which indicates a docked situation. System card 28 on the other side of mux 24 reads the I/O device and signals that it is being docked. The system pulls mux 24 and lets the cpu in notebook have control over the released resources in docking station 92. As discussed previously, one way of accomplishing this is to write simple code to the cpu in system card 28 that says "if there is an input (interrupt) from the mux (which you could hook to external interrupt 1 or 2 off of the SMI interrupt on the CPU), the codes see this and says, "read the status--docked or undocked". If docked, the docking station turns over the resources, the CPU quits running on the PCI bus 14 (it doesn't issue any PCI cycles) and throws the mux so that the PCI bus is being driven by the external master and not the CPU in the system card 28. When the mux goes back, then the CPU in system card 28 starts driving the internal PCI bus 14. Thus, in a docked situation with a notebook, notebook may be given partial or full control over the resources of system card 28 and additional cards 30.

Detailed Description Text (373):
In yet other embodiments of the invention, illustrated in FIGS. 410, 414, 418 and 422, at least one additional card slot (three actually shown) are added to the docking stations of FIGS. 408, 412, 416 and 420, respectively. Each card slot 26 is coupled to mux 24. Each additional card slot is coupleable to a card 30. Each card 30 is treated as a system resource. When a notebook docks to docking station 94 it will pull the pin high on the mux which indicates a docked situation. System card 28 on the other side of mux 24 reads the I/O device and signals that it is being docked. The system pulls mux 24 and lets the cpu in notebook have control over the released resources in docking station 94. As discussed previously, one way of accomplishing this is to write simple code to the cpu in system card 28 that says "if there is an input (interrupt) from the mux (which you could hook to external interrupt 1 or 2 off of the SMI interrupt on the CPU), the codes see this and says, "read the status--docked or undocked". If docked, the docking station turns over the resources, the CPU quits running on the PCI bus 14 (it doesn't issue any PCI cycles) and throws the mux so that the PCI bus is being driven by the external master and not the CPU in the system card 28. When the mux goes back, then the CPU in system card 28 starts driving the internal PCI bus 14. Thus, in a docked situation with a notebook, notebook may be given partial or full control over the resources of system card 28 and additional cards 30.

Detailed Description Text (375):
In still other embodiments of the invention, illustrated in FIGS. 411, 415, 419, and 423, at least one additional card slot (three actually shown) are added to the docking stations of FIGS. 409, 413, 417, and 421, respectively. Each card slot 26 is coupled to mux 24. Each additional card slot is coupleable to a card 30. Each card 30 is treated as a system resource. When a notebook having at least one corresponding interface docks to docking station 94, it will pull the pin high on the mux which indicates a docked situation. System card 28 on the other side of mux 24 reads the I/O device and signals that it is being docked. The system pulls mux 24 and lets the cpu in notebook have control over the released resources in docking station 94. As discussed previously, one way of accomplishing this is to write simple code to the cpu in system card 28 that says "if there is an input (interrupt) from the mux (which you could hook to external interrupt 1 or 2 off of the SMI interrupt on the CPU), the codes see this and says, "read the status--docked or undocked". If docked, the docking station turns over the resources, the CPU quits running on the PCI bus 14 (it doesn't issue any PCI cycles) and throws the mux so that the PCI bus is being driven by the external master and not the CPU in the system card 28. When the mux goes back, then the CPU in system card 28 starts driving the internal PCI bus 14. Thus, in a docked situation with a notebook, notebook may be given partial or full control over the resources of system card 28 and additional cards 30.

Detailed Description Text (383):
In yet other embodiments of the invention, illustrated in FIG. 426, at least one

additional card slot (three actually shown) are added to the docking station of FIG. 424. Each card slot 26 is coupled to mux 24. Each additional card slot is coupleable to a card 30. Each card 30 is treated as a system resource. When a notebook docks to docking station 96 it will pull the pin high on the mux which indicates a docked situation. System card 28 on the other side of mux 24 reads the I/O device and signals that it is being docked. The system pulls mux 24 and lets the cpu in notebook have control over the released resources in docking station 96. As discussed previously, one way of accomplishing this is to write simple code to the cpu in system card 28 that says "if there is an input (interrupt) from the mux (which you could hook to external interrupt 1 or 2 off of the SMI interrupt on the CPU), the codes see this and says, "read the status--docked or undocked". If docked, the docking station turns over the resources, the CPU quits running on the PCI bus 14 (it doesn't issue any PCI cycles) and throws the mux so that the PCI bus is being driven by the external master and not the CPU in the system card 28. When the mux goes back, then the CPU in system card 28 starts driving the internal PCI bus 14. Thus, in a docked situation with a notebook, notebook may be given partial or full control over the resources of system card 28 and additional cards 30.

Detailed Description Text (384):
In still embodiments of the invention, illustrated in FIG. 427, at least one additional card slot (three actually shown) are added to the docking station of FIG. 425. Each card slot 26 is coupled to mux 24. Each additional card slot is coupleable to a card 30. Each card 30 is treated as a system resource. When a notebook having at least one corresponding interface docks to docking station 96, it will pull the pin high on the mux which indicates a docked situation. System card 28 on the other side of mux 24 reads the I/O device and signals that it is being docked. The system pulls mux 24 and lets the cpu in notebook have control over the released resources in docking station 94. As discussed previously, one way of accomplishing this is to write simple code to the cpu in system card 28 that says "if there is an input (interrupt) from the mux (which you could hook to external interrupt 1 or 2 off of the SMI interrupt on the CPU), the codes see this and says, "read the status--docked or undocked". If docked, the docking station turns over the resources, the CPU quits running on the PCI bus 14 (it doesn't issue any PCI cycles) and throws the mux so that the PCI bus is being driven by the external master and not the CPU in the system card 28. When the mux goes back, then the CPU in system card 28 starts driving the internal PCI bus 14. Thus, in a docked situation with a notebook, notebook may be given partial or full control over the resources of system card 28 and additional cards 30.

WEST

☐ Generate Collection | Print

L6: Entry 4 of 33 File: USPT Nov 21, 2000

US-PAT-NO: 6151646
DOCUMENT-IDENTIFIER: US 6151646 A

TITLE: System for resources under control of docking station when stand alone and resources under control of central processing unit of portable computer when docked

DATE-ISSUED: November 21, 2000

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Watts; LaVaughn F.	Temple	TX		
Linn; John C.	Richarson	TX		

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
Texas Instruments Incorporated	Dallas	TX			02

APPL-NO: 09/ 304935 [PALM]
DATE FILED: May 4, 1999

PARENT-CASE:
This application is a continuation of Ser. No. 08/651,165 filed May 2, 1996 .

INT-CL: [07] G06 F 9/00

US-CL-ISSUED: 710/72; 710/38, 710/8
US-CL-CURRENT: 710/72; 710/38, 710/8

FIELD-OF-SEARCH: 710/69, 710/72, 710/38, 710/8, 714/25, 364/400.01, 364/708.1, 364/280.2, 3/892

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

Search Selected | Search ALL

	PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
☐	5931929	August 1999	Tran et al.	710/69
☐	5968187	October 1999	Robinson	714/25
☐	6007228	December 1999	Agarwal et al.	364/400.01

ART-UNIT: 272

PRIMARY-EXAMINER: Lee; Thomas C.

ASSISTANT-EXAMINER: Kim; Harold

ATTY-AGENT-FIRM: Neerings; Ronald O. Brady, III; Wade James Telecky, Jr.; Frederick J.

ABSTRACT:

The described embodiments of the present invention provide a computer docking station (12, 32, 58, 68, 76, 84, 90, 92, 94, 96) that can have its functionality reconfigured by a docked portable personal computer (10, 38, 62, 66, 74, 82). In at least one embodiment of the invention, the computer docking station is configured as a stand alone computer prior to docking with a portable computer, may have its functionality reconfigured when docked to the portable computer, and reconfigures itself to be a stand alone computer when undocked from the portable computer. In one embodiment of the invention, docking station resources are placed under the control of a docked portable computer. In another embodiment of the invention, docked portable computer resources are placed under the control of the docking station. The invention contemplates docking via direct connection, radio requency "RF" communications, infrared "IR" communications, 1394 high performance serial bus communications, or card bus communications, and/or combinations of one or more of these communications techniques.

12 Claims, 427 Drawing figures

WEST

Generate Collection Print

L6: Entry 7 of 33 File: USPT Feb 8, 2000

DOCUMENT-IDENTIFIER: US 6023587 A
TITLE: System for resources under control of docking station when standalone and resources under control of central processing unit of portable computer when docked

Abstract Text (1):
The described embodiments of the present invention provide a computer docking station (12, 32, 58, 68, 76, 84, 90, 92, 94, 96) that can have its functionality reconfigured by a docked portable personal computer (10, 38, 62, 66, 74, 82). In at least one embodiment of the invention, the computer docking station is configured as a stand alone computer prior to docking with a portable computer, may have its functionality reconfigured when docked to the portable computer, and reconfigures itself to be a stand alone computer when undocked from the portable computer. In one embodiment of the invention, docking station resources are placed under the control of a docked portable computer. In another embodiment of the invention, docked portable computer resources are placed under the control of the docking station. The invention contemplates docking via direct connection, radio requency "RF" communications, infrared "IR" communications, 1394 high performance serial bus communications, or card bus communications, and/or combinations of one or more of these communications techniques.

Detailed Description Text (37):
MUX 24 is fairly straight forward. It is really a switch with data lines that signal whether or not there is a docked notebook computer to take control over some of the docking station resources. One method pulls a pin high to provide this function. When notebook 10 hard docks to docking station 12, it will pull the pin high on the mux which indicates a docked situation. System card 28 on the other side of mux 24 reads the I/O device and signals that it is being docked. The system pulls mux 24 and lets the cpu in notebook 10 have control over the released resources in docking station 12. One way of accomplishing this is to write simple code to the cpu in system card 28 that says "if there is an input (interrupt) from the mux (which you could hook to external interrupt 1 or 2 off of the SMI interrupt on the CPU), tile codes see this and says, "read the status--docked or undocked". If docked, you turn over the devices, the CPU quits running on the PCI bus (it doesn't issue any PCI cycles) and throws the mux so that the PCI bus is being driven by the external master and not the CPU in the card. When the mux goes back, then the CPU starts driving the internal PCI bus. Thus, in a docked situation with a notebook, notebook may be given partial or full control over the resources of system card 28.

Detailed Description Text (209):
In yet another embodiment of the invention, illustrated in FIG. 264, at least one additional card slot (three actually shown) ise added to the docking station of FIG. 3. Each additional card slot 26 is coupled to mux 24. Each additional card slot is coupleable to a card 30. Each card 30 contains at least one resource that is an addition to the resources available on system card 28. Being that cards 30 couple to mux 24, the cpu in the docking station (on system card 28 in this embodiment), controls the resources of cards 30 when the docking station is "undocked". When notebook 10 docks to docking station 12, as illustrated in FIG. 265, it will pull the pin high on the mux which indicates a docked situation. System card 28 on the other side of mux 24 reads the I/O device and signals that it is being docked. The system pulls mux 24 and lets the cpu in notebook 10 have control over the released resources in docking station 12, be they resources on system card 28 and/or card(s) 30. As discussed previously, one way of accomplishing this is to write simple code

to the cpu in system card 28 that says "if there is an input (interrupt) from the mux (which you could hook to external interrupt 1 or 2 off of the SMI interrupt on the CPU), the codes see this and says, " read the status--docked or undocked". If docked, the docking station turns over the resources, the CPU quits running on the PCI bus 14 (it doesn't issue any PCI cycles) and throws the mux so that the PCI bus is being driven by the external master and not the CPU in the system card 28. When the mux goes back, then the CPU in system card 28 starts driving the internal PCI bus 14. Thus, in a docked situation with a notebook, notebook may be given partial or full control over the resources of system card 28 and additional cards 30.

Detailed Description Text (277):
In yet another embodiment of the invention, illustrated in FIG. 327, at least one additional card slot (three actually shown) are added to the docking station of FIG. 319. Each card slot 26 is coupled to mux 24. Each additional card slot is coupleable to a card 30. Each card 30 is treated as a system resource. When notebook 38 docks to docking station 32, as illustrated in FIG. 328, it will pull the pin high on the mux which indicates a docked situation. System card 28 on the other side of mux 24 reads the I/O device and signals that it is being docked. The system pulls mux 24 and lets the cpu in notebook 10 have control over the released resources in docking station 32. As discussed previously, one way of accomplishing this is to write simple code to the cpu in system card 28 that says "if there is an input (interrupt) from the mux (which you could hook to external interrupt 1 or 2 off of the SMI interrupt on the CPU), the codes see this and says, "read the status--docked or undocked". If docked, the docking station turns over the resources, the CPU quits running on the PCI bus 14 (it doesn't issue any PCI cycles) and throws the mux so that the PCI bus is being driven by the external master and not the CPU in the system card 28. When the mux goes back, then the CPU in system card 28 starts driving the internal PCI bus 14. Thus, in a docked situation with a notebook, notebook may be given partial or full control over the resources of system card 28 and additional cards 30.

Detailed Description Text (279):
In yet another embodiment of the invention, illustrated in FIG. 329, at least one additional card slot (three actually shown) are added to the docking station of FIG. 325. Each card slot 26 is coupled to mux 24. Each additional card slot is coupleable to a card 30. Each card 30 is treated as a system resource. When notebook 38 docks to docking station 32, illustrated in FIG. 330, it will pull the pin high on the mux which indicates a docked situation. System card 28 on the other side of mux 24 reads the I/O device and signals that it is being docked. The system pulls mux 24 and lets the cpu in notebook 10 have control over the released resources in docking station 12. As discussed previously, one way of accomplishing this is to write simple code to the cpu in system card 28 that says "if there is an input (interrupt) from the mux (which you could hook to external interrupt 1 or 2 off of the SMI interrupt on the CPU), the codes see this and says, "read the status--docked or undocked". If docked, the docking station turns over the resources, the CPU quits running on the PCI bus 14 (it doesn't issue any PCI cycles) and throws the mux so that the PCI bus is being driven by the external master and not the CPU in the system card 28. When the mux goes back, then the CPU in system card 28 starts driving the internal PCI bus 14. Thus, in a docked situation with a notebook, notebook may be given partial or full control over the resources of system card 28 and additional cards 30.

Detailed Description Text (299):
In yet another embodiment of the invention, illustrated in FIG. 337, at least one additional card slot (three actually shown) are added to the docking station of FIG. 331. Each card slot 26 is coupled to mux 24. Each additional card slot is coupleable to a card 30. Each card 30 is treated as a system resource. When notebook 62 docks to docking station 58, as illustrated in FIG. 338, it will pull the pin high on the mux which indicates a docked situation. System card 28 on the other side of mux 24 reads the I/O device and signals that it is being docked. The system pulls mux 24 and lets the cpu in notebook 10 have control over the released resources in docking station 58. As discussed previously, one way of accomplishing this is to write simple code to the cpu in system card 28 that says "if there is an input (interrupt) from the mux (which you could hook to external interrupt 1 or 2 off of the SMI interrupt on the CPU), the codes see this and says, "read the status--docked or undocked". If docked, the docking station turns over the resources, the CPU quits

running on the PCI bus 14 (it doesn't issue any PCI cycles) and throws the mux so that the PCI bus is being driven by the external master and not the CPU in the system card 28. When the mux goes back, then the CPU in system card 28 starts driving the internal PCI bus 14. Thus, in a docked situation with a notebook, notebook may be given partial or full control over the resources of system card 28 and additional cards 30.

Detailed Description Text (301):
In yet another embodiment of the invention, illustrated in FIG. 339, at least one additional card slot (three actually shown) are added to the docking station of FIG. 335. Each card slot 26 is coupled to mux 24. Each additional card slot is coupleable to a card 30. Each card 30 is treated as a system resource. When notebook 38 docks to docking station 58, illustrated in FIG. 336, it will pull the pin high on the mux which indicates a docked situation. System card 28 on the other side of mux 24 reads the I/O device and signals that it is being docked. The system pulls mux 24 and lets the cpu in notebook 10 have control over the released resources in docking station 58. As discussed previously, one way of accomplishing this is to write simple code to the cpu in system card 28 that says "if there is an input (interrupt) from the mux (which you could hook to external interrupt 1 or 2 off of the SMI interrupt on the CPU), the codes see this and says, "read the status--docked or undocked". If docked, the docking station turns over the resources, the CPU quits running on the PCI bus 14 (it doesn't issue any PCI cycles) and throws the mux so that the PCI bus is being driven by the external master and not the CPU in the system card 28. When the mux goes back, then the CPU in system card 28 starts driving the internal PCI bus 14. Thus, in a docked situation with a notebook, notebook may be given partial or full control over the resources of system card 28 and additional cards 30.

Detailed Description Text (313):
In yet another embodiment of the invention, illustrated in FIG. 346, at least one additional card slot (three actually shown) are added to the docking station of FIG. 341. Each card slot 26 is coupled to mux 24. Each additional card slot is coupleable to a card 30. Each card 30 is treated as a system resource. When notebook 66 docks to docking station 68, as illustrated in FIG. 347, it will pull the pin high on the mux which indicates a docked situation. System card 28 on the other side of mux 24 reads the I/O device and signals that it is being docked. The system pulls mux 24 and lets the cpu in notebook 66 have control over the released resources in docking station 68. As discussed previously, one way of accomplishing this is to write simple code to the cpu in system card 28 that says "if there is an input (interrupt) from the mux (which you could hook to external interrupt 1 or 2 off of the SMI interrupt on the CPU), the codes see this and says, "read the status--docked or undocked". If docked, the docking station turns over the resources, the CPU quits running on the PCI bus 14 (it doesn't issue any PCI cycles) and throws the mux so that the PCI bus is being driven by the external master and not the CPU in the system card 28. When the mux goes back, then the CPU in system card 28 starts driving the internal PCI bus 14. Thus, in a docked situation with a notebook, notebook may be given partial or full control over the resources of system card 28 and additional cards 30.

Detailed Description Text (315):
In yet another embodiment of the invention, illustrated in FIG. 348, at least one additional card slot (three actually shown) are added to the docking station of FIG. 344. Each card slot 26 is coupled to mux 24. Each additional card slot is coupleable to a card 30. Each card 30 is treated as a system resource. When notebook 66 docks to docking station 68, illustrated in FIG. 349, it will pull the pin high on the mux which indicates a docked situation. System card 28 on the other side of mux 24 reads the I/O device and signals that it is being docked. The system pulls mux 24 and lets the cpu in notebook 66 have control over the released resources in docking station 68. As discussed previously, one way of accomplishing this is to write simple code to the cpu in system card 28 that says "if there is an input (interrupt) from the mux (which you could hook to external interrupt 1 or 2 off of the SMI interrupt on the CPU), the codes see this and says, "read the status--docked or undocked". If docked, the docking station turns over the resources, the CPU quits running on the PCI bus 14 (it doesn't issue any PCI cycles) and throws the mux so that the PCI bus is being driven by the external master and not the CPU in the system card 28. When the mux goes back, then the CPU in system card 28 starts driving the internal PCI

bus 14. Thus, in a docked situation with a notebook, notebook may be given partial or full control over the resources of system card 28 and additional cards 30.

Detailed Description Text (323):
In yet another embodiment of the invention, illustrated in FIG. 355, at least one additional card slot (three actually shown) are added to the docking station of FIG. 350. Each card slot 26 is coupled to mux 24. Each additional card slot is coupleable to a card 30. Each card 30 is treated as a system resource. When notebook 74 docks to docking station 76, as illustrated in FIG. 356, it will pull the pin high on the mux which indicates a docked situation. System card 28 on the other side of mux 24 reads the I/O device and signals that it is being docked. The system pulls mux 24 and lets the cpu in notebook 74 have control over the released resources in docking station 76. As discussed previously, one way of accomplishing this is to write simple code to the cpu in system card 28 that says "if there is an input (interrupt) from the mux (which you could hook to external interrupt 1 or 2 off of the SMI interrupt on the CPU), the codes see this and says, "read the status--docked or undocked". If docked, the docking station turns over the resources, the CPU quits running on the PCI bus 14 (it doesn't issue any PCI cycles) and throws the mux so that the PCI bus is being driven by the external master and not the CPU in the system card 28. When the mux goes back, then the CPU in system card 28 starts driving the internal PCI bus 14. Thus, in a docked situation with a notebook, notebook may be given partial or full control over the resources of system card 28 and additional cards 30.

Detailed Description Text (325):
In yet another embodiment of the invention, illustrated in FIG. 357, at least one additional card slot (three actually shown) are added to the docking station of FIG. 353. Each card slot 26 is coupled to mux 24. Each additional card slot is coupleable to a card 30. Each card 30 is treated as a system resource. When notebook 74 docks to docking station 76, illustrated in FIG. 358, it will pull the pin high on the mux which indicates a docked situation. System card 28 on the other side of mux 24 reads the I/O device and signals that it is being docked. The system pulls mux 24 and lets the cpu in notebook 74 have control over the released resources in docking station 76. As discussed previously, one way of accomplishing this is to write simple code to the cpu in system card 28 that says "if there is an input (interrupt) from the mux (which you could hook to external interrupt 1 or 2 off of the SMI interrupt on the CPU), the codes see this and says, "read the status--docked or undocked". If docked, the docking station turns over the resources, the CPU quits running on the PCI bus 14 (it doesn't issue any PCI cycles) and throws the mux so that the PCI bus is being driven by the external master and not the CPU in the system card 28. When the mux goes back, then the CPU in system card 28 starts driving the internal PCI bus 14. Thus, in a docked situation with a notebook, notebook may be given partial or full control over the resources of system card 28 and additional cards 30.

Detailed Description Text (334):
In yet another embodiment of the invention, illustrated in FIG. 364, at least one additional card slot (three actually shown) are added to the docking station of FIG. 359. Each card slot 26 is coupled to mux 24. Each additional card slot is coupleable to a card 30. Each card 30 is treated as a system resource. When notebook 82 docks to docking station 84, as illustrated in FIG. 365, it will pull the pin high on the mux which indicates a docked situation. System card 28 on the other side of mux 24 reads the I/O device and signals that it is being docked. The system pulls mux 24 and lets the cpu in notebook 10 have control over the released resources in docking station 84. As discussed previously, one way of accomplishing this is to write simple code to the cpu in system card 28 that says "if there is an input (interrupt) from the mux (which you could hook to external interrupt 1 or 2 off of the SMI interrupt on the CPU), the codes see this and says, "read the status--docked or undocked". If docked, the docking station turns over the resources, the CPU quits running on the PCI bus 14 (it doesn't issue any PCI cycles) and throws the mux so that the PCI bus is being driven by the external master and not the CPU in the system card 28. When the mux goes back, then the CPU in system card 28 starts driving the internal PCI bus 14. Thus, in a docked situation with a notebook, notebook may be given partial or full control over the resources of system card 28 and additional cards 30.

Detailed Description Text (336):
In yet another embodiment of the invention, illustrated in FIG. 366, at least one additional card slot (three actually shown) are added to the docking station of FIG. 362. Each card slot 26 is coupled to mux 24. Each additional card slot is coupleable to a card 30. Each card 30 is treated as a system resource. When notebook 82 docks to docking station 84, illustrated in FIG. 367, it will pull the pin high on the mux which indicates a docked situation. System card 28 on the other side of mux 24 reads the I/O device and signals that it is being docked. The system pulls mux 24 and lets the cpu in notebook 10 have control over the released resources in docking station 58. As discussed previously, one way of accomplishing this is to write simple code to the cpu in system card 28 that says "if there is an input (interrupt) from the mux (which you could hook to external interrupt 1 or 2 off of the SMI interrupt on the CPU), the codes see this and says, "read the status--docked or undocked". If docked, the docking station turns over the resources, the CPU quits running on the PCI bus 14 (it doesn't issue any PCI cycles) and throws the mux so that the PCI bus is being driven by the external master and not the CPU in the system card 28. When the mux goes back, then the CPU in system card 28 starts driving the internal PCI bus 14. Thus, in a docked situation with a notebook, notebook may be given partial or full control over the resources of system card 28 and additional cards 30.

Detailed Description Text (348):
In yet other embodiments of the invention, illustrated in FIGS. 370, 374, 378 and 382, at least one additional card slot (three actually shown) are added to the docking stations of FIGS. 368, 372, 376 and 380, respectively. Each card slot 26 is coupled to mux 24. Each additional card slot is coupleable to a card 30. Each card 30 is treated as a system resource. When a notebook docks to docking station 90 it will pull the pin high on the mux which indicates a docked situation. System card 28 on the other side of mux 24 reads the I/O device and signals that it is being docked. The system pulls mux 24 and lets the cpu in notebook have control over the released resources in docking station 90. As discussed previously, one way of accomplishing this is to write simple code to the cpu in system card 28 that says "if there is an input (interrupt) from the mux (which you could hook to external interrupt 1 or 2 off of the SMI interrupt on the CPU), the codes see this and says, "read the status --docked or undocked". If docked, the docking station turns over the resources, the CPU quits running on the PCI bus 14 (it doesn't issue any PCI cycles) and throws the mux so that the PCI bus is being driven by the external master and not the CPU in the system card 28. When the mux goes back, then the CPU in system card 28 starts driving the internal PCI bus 14. Thus, in a docked situation with a notebook, notebook may be given partial or full control over the resources of system card 28 and additional cards 30.

Detailed Description Text (355):
In yet other embodiments of the invention, illustrated in FIGS. 386, 390, 394 and 398, at least one additional card slot (three actually shown) are added to the docking stations of FIGS. 384, 388, 392 and 396, respectively. Each card slot 26 is coupled to mux 24. Each additional card slot is coupleable to a card 30. Each card 30 is treated as a system resource. When a notebook docks to docking station 92 it will pull the pin high on the mux which indicates a docked situation. System card 28 on the other side of mux 24 reads the I/O device and signals that it is being docked. The system pulls mux 24 and lets the cpu in notebook have control over the released resources in docking station 92. As discussed previously, one way of accomplishing this is to write simple code to the cpu in system card 28 that says "if there is an input (interrupt) from the mux (which you could hook to external interrupt 1 or 2 off of the SMI interrupt on the CPU), the codes see this and says, "read the status--docked or undocked". If docked, the docking station turns over the resources, the CPU quits running on the PCI bus 14 (it doesn't issue any PCI cycles) and throws the mux so that the PCI bus is being driven by the external master and not the CPU in the system card 28. When the mux goes back, then the CPU in system card 28 starts driving the internal PCI bus 14. Thus, in a docked situation with a notebook, notebook may be given partial or full control over the resources of system card 28 and additional cards 30.

Detailed Description Text (357):
In still other embodiments of the invention, illustrated in FIGS. 387, 391, 395, and 399, at least one additional card slot (three actually shown) are added to the

docking station of FIG. 385. Each card slot 26 is coupled to mux 24. Each additional card slot is coupleable to a card 30. Each card 30 is treated as a system resource. When a notebook having at least one corresponding interface docks to docking station 92, it will pull the pin high on the mux which indicates a docked situation. System card 28 on the other side of mux 24 reads the I/O device and signals that it is being docked. The system pulls mux 24 and lets the cpu in notebook have control over the released resources in docking station 92. As discussed previously, one way of accomplishing this is to write simple code to the cpu in system card 28 that says "if there is an input (interrupt) from the mux (which you could hook to external interrupt 1 or 2 off of the SMI interrupt on the CPU), the codes see this and says, "read the status--docked or undocked". If docked, the docking station turns over the resources, the CPU quits running on the PCI bus 14 (it doesn't issue any PCI cycles) and throws the mux so that the PCI bus is being driven by the external master and not the CPU in the system card 28. When the mux goes back, then the CPU in system card 28 starts driving the internal PCI bus 14. Thus, in a docked situation with a notebook, notebook may be given partial or full control over the resources of system card 28 and additional cards 30.

Detailed Description Text (365):
In yet other embodiments of the invention, illustrated in FIGS. 410, 414, 418 and 422, at least one additional card slot (three actually shown) are added to the docking stations of FIGS. 408, 412, 416 and 420, respectively. Each card slot 26 is coupled to mux 24. Each additional card slot is coupleable to a card 30. Each card 30 is treated as a system resource. When a notebook docks to docking station 94 it will pull the pin high on the mux which indicates a docked situation. System card 28 on the other side of mux 24 reads the I/O device and signals that it is being docked. The system pulls mux 24 and lets the cpu in notebook have control over the released resources in docking station 94. As discussed previously, one way of accomplishing this is to write simple code to the cpu in system card 28 that says "if there is an input (interrupt) from the mux (which you could hook to external interrupt 1 or 2 off of the SMI interrupt on the CPU), the codes see this and says, "read the status--docked or undocked". If docked, the docking station turns over the resources, the CPU quits running on the PCI bus 14 (it doesn't issue any PCI cycles) and throws the mux so that the PCI bus is being driven by the external master and not the CPU in the system card 28. When the mux goes back, then the CPU in system card 28 starts driving the internal PCI bus 14. Thus, in a docked situation with a notebook, notebook may be given partial or full control over the resources of system card 28 and additional cards 30.

Detailed Description Text (367):
In still other embodiments of the invention, illustrated in FIGS. 411, 415, 419, and 423, at least one additional card slot (three actually shown) are added to the docking stations of FIGS. 409, 413, 417, and 421, respectively. Each card slot 26 is coupled to mux 24. Each additional card slot is coupleable to a card 30. Each card 30 is treated as a system resource. When a notebook having at least one corresponding interface docks to docking station 94, it will pull the pin high on the mux which indicates a docked situation. System card 28 on the other side of mux 24 reads the I/O device and signals that it is being docked. The system pulls mux 24 and lets the cpu in notebook have control over the released resources in docking station 94. As discussed previously, one way of accomplishing this is to write simple code to the cpu in system card 28 that says "if there is an input (interrupt) from the mux (which you could hook to external interrupt 1 or 2 off of the SMI interrupt on the CPU), the codes see this and says, "read the status--docked or undocked". If docked, the docking station turns over the resources, the CPU quits running on the PCI bus 14 (it doesn't issue any PCI cycles) and throws the mux so that the PCI bus is being driven by the external master and not the CPU in the system card 28. When the mux goes back, then the CPU in system card 28 starts driving the internal PCI bus 14. Thus, in a docked situation with a notebook, notebook may be given partial or full control over the resources of system card 28 and additional cards 30.

Detailed Description Text (375):
In yet other embodiments of the invention, illustrated in FIG. 426, at least one additional card slot (three actually shown) are added to the docking station of FIG. 424. Each card slot 26 is coupled to mux 24. Each additional card slot is coupleable

to a card 30. Each card 30 is treated as a system resource. When a notebook docks to docking station 96 it will pull the pin high on the mux which indicates a docked situation. System card 28 on the other side of mux 24 reads the I/O device and signals that it is being docked. The system pulls mux 24 and lets the cpu in notebook have control over the released resources in docking station 96. As discussed previously, one way of accomplishing this is to write simple code to the cpu in system card 28 that says "if there is an input (interrupt) from the mux (which you could hook to external interrupt 1 or 2 off of the SMI interrupt on the CPU), the codes see this and says, "read the status--docked or undocked". If docked, the docking station turns over the resources, the CPU quits running on the PCI bus 14 (it doesn't issue any PCI cycles) and throws the mux so that the PCI bus is being driven by the external master and not the CPU in the system card 28. When the mux goes back, then the CPU in system card 28 starts driving the internal PCI bus 14. Thus, in a docked situation with a notebook, notebook may be given partial or full control over the resources of system card 28 and additional cards 30.

Detailed Description Text (376):
In still other embodiments of the invention, illustrated in FIG. 427, at least one additional card slot (three actually shown) are added to the docking station of FIG. 425. Each card slot 26 is coupled to mux 24. Each additional card slot is coupleable to a card 30. Each card 30 is treated as a system resource. When a notebook having at least one corresponding interface docks to docking station 96, it will pull the pin high on the mux which indicates a docked situation. System card 28 on the other side of mux 24 reads the I/O device and signals that it is being docked. The system pulls mux 24 and lets the cpu in notebook have control over the released resources in docking station 94. As discussed previously, one way of accomplishing this is to write simple code to the cpu in system card 28 that says "if there is an input (interrupt) from the mux (which you could hook to external interrupt I or 2 off of the SMI interrupt on the CPU), the codes see this and says, "read the status--docked or undocked". If docked, the docking station turns over the resources, the CPU quits running on the PCI bus 14 (it doesn't issue any PCI cycles) and throws the mux so that the PCI bus is being driven by the external master and not the CPU in the system card 28. When the mux goes back, then the CPU in system card 28 starts driving the internal PCI bus 14. Thus, in a docked situation with a notebook, notebook may be given partial or full control over the resources of system card 28 and additional cards 30.

WEST

☐ Generate Collection | Print

L6: Entry 7 of 33 File: USPT Feb 8, 2000

US-PAT-NO: 6023587
DOCUMENT-IDENTIFIER: US 6023587 A

TITLE: System for resources under control of docking station when standalone and resources under control of central processing unit of portable computer when docked

DATE-ISSUED: February 8, 2000

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Watts, Jr.; LaVaughn F.	Temple	TX		
Linn; John C.	Richarson	TX		

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
Texas Instruments Incorporated	Dallas	TX			02

APPL-NO: 08/ 651165 [PALM]
DATE FILED: May 2, 1996

INT-CL: [06] G06 F 9/00

US-CL-ISSUED: 395/892; 395/858, 395/828
US-CL-CURRENT: 710/72

FIELD-OF-SEARCH: 364/708.1, 364/280.2, 395/281, 395/306, 395/500, 395/892, 395/858, 395/828, 3/892

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

Search Selected | Search ALL

	PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
☐	5265238	November 1993	Canova, Jr. et al.	395/500
☐	5297272	March 1994	Lu et al.	395/500
☐	5444869	August 1995	Strickin et al.	455/89
☐	5463742	October 1995	Kobayashi	395/281
☐	5515514	May 1996	Dhuey et al.	395/282
☐	5579528	November 1996	Register	395/671
☐	5598539	January 1997	Gephardt et al.	395/281
☐	5625829	April 1997	Gephardt et al.	395/800
☐	5668977	September 1997	Swanstrom et al.	395/500
☐	5671366	September 1997	Niwa et al.	395/281

OTHER PUBLICATIONS

Apple Computer, Macintosh Family Hardware Reference, 1988, pp. 16-116-117.

ART-UNIT: 272

PRIMARY-EXAMINER: Lee; Thomas C.

ASSISTANT-EXAMINER: Chen; Anderson I.

ATTY-AGENT-FIRM: Neerings; Ronald O. Holland; Robby T. Donaldson; Richard L.

ABSTRACT:

The described embodiments of the present invention provide a computer docking station (12, 32, 58, 68, 76, 84, 90, 92, 94, 96) that can have its functionality reconfigured by a docked portable personal computer (10, 38, 62, 66, 74, 82). In at least one embodiment of the invention, the computer docking station is configured as a stand alone computer prior to docking with a portable computer, may have its functionality reconfigured when docked to the portable computer, and reconfigures itself to be a stand alone computer when undocked from the portable computer. In one embodiment of the invention, docking station resources are placed under the control of a docked portable computer. In another embodiment of the invention, docked portable computer resources are placed under the control of the docking station. The invention contemplates docking via direct connection, radio requency "RF" communications, infrared "IR" communications, 1394 high performance serial bus communications, or card bus communications, and/or combinations of one or more of these communications techniques.

28 Claims, 427 Drawing figures

WEST

Generate Collection Print

L6: Entry 13 of 33 File: USPT Feb 2, 1999

DOCUMENT-IDENTIFIER: US 5867406 A
TITLE: Docking device for a portable computer and a method for docking a portable computer to the docking device

Abstract Text (1):
A docking device for a portable computer includes a body adapted to receive a portable computer. The body encloses a number of electronic components adapted to be connected to a portable computer via a number of signal lines. A connector is provided in the body which mechanically connects the number of signal lines within the body to a corresponding number of signal lines within the portable computer. The docking device also includes a signal connection means interposed between the portable computer and the electronic components for selectively electrically interconnecting the particular ones of the signal lines within the body to corresponding signal lines among the number of signal lines within the portable computer in response to an operating state of said portable computer. A preferred embodiment of the docking device further includes a control means for controlling the selective electrical interconnection of the particular ones of the number of signal lines by the signal connection means. In this preferred embodiment, the control means electrically interconnects the particular ones of the number of signal lines within the body to corresponding signal lines within the portable computer in response to an acknowledgment by the portable computer of a request by the control means to electrically interconnect the particular ones of the number of signal lines.

Detailed Description Text (37):
Next, at block 204 OS/utility 72 determines by software whether or not to actually perform docking. For instance, if the system configuration of docking device 60 conflicts with the resources already installed in host 50, the docking cannot be performed. In this case, the process proceeds to block 206, where an eject command is sent to docking device 60 to completely undock host 50. Thereafter, host 50 is ejected at block 208 and the process terminates at block 210. If, however, at block 204 OS 72 determines that partial docking can be performed, the process proceeds to block 212, which depicts abandoning the electrical connection of main bus 33 and maintaining the connection between host 50 and ports within docking device 60. In addition, the OS 72 of host 50 determines the system configuration of docking device 60 by reading the content of the EEPROM additionally provided in the CPU 42 or by reading a prestored correspondence table including the identification number and the system configuration of docking device 60.

Detailed Description Text (47):
Referring now to FIG. 12, there is depicted the undocking process subsequent to off-page connector T of FIG. 11. The process proceeds from on-page connector T to block 600, where if host 50 is in an ordinary operating state, the interface circuit 41 issues an SMI to host 50. The interrupt handler 24 detects and reports the occurrence of the SMI to BIOS 73. When BIOS 73 detects that the SMI has been issued by docking device 60, host 50 reads the content (`ABOUT.sub.-- TO.sub.-- UNDOCK`) of the output register in the interface circuit 41 and reports to OS 72 that host 50 and docking device 60 are about to be undocked at block 602. Then, OS 72 sends a command to docking device 60 to isolate the bus at block 604. More particularly, the command is transmitted by writing a record `BUS.sub.-- DISCONNECT` to the input register in the interface circuit 41 and setting the IBF. Setting the IBF causes an interrupt to occur in CPU 42. Thereafter, CPU 42 sets the busy flag to process

`BUS.sub.-- DISCONNECT`. Then, the CPU 42 writes a code "Acknowledge" indicating acceptance of the command to the output register and sets the OBF. BIOS 73 polls the OBF and reads the content of the output register in response to setting the OBF. BIOS 72 confirms that the command has been accepted by resetting the OBF. Then, at block 606 CPU 42 determines through the resetting of the OBF that the command has been accepted by host 50 and causes the interface circuit 41 to isolate main bus 33 and subbus 34. Like connection of main bus 33, isolation is performed at the start of the refresh cycle. After the isolation of the bus signals 33 and 34, host 50 may be ejected from docking device 60 by a mechanical operation. Thereafter, the process terminates at block 608.

WEST

☐ Generate Collection | Print

L6: Entry 13 of 33 File: USPT Feb 2, 1999

US-PAT-NO: 5867406
DOCUMENT-IDENTIFIER: US 5867406 A

TITLE: Docking device for a portable computer and a method for docking a portable computer to the docking device

DATE-ISSUED: February 2, 1999

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Yanagisawa; Takashi	Yokohama			JP

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
International Business Machines Corp.	Armonk	NY			02

APPL-NO: 08/ 416398 [PALM]
DATE FILED: April 6, 1995

FOREIGN-APPL-PRIORITY-DATA:

COUNTRY	APPL-NO	APPL-DATE
JP	6-134124	June 16, 1994

INT-CL: [06] G06 F 7/50

US-CL-ISSUED: 364/708.1; 361/683, 361/729, 361/733
US-CL-CURRENT: 708/140; 361/683, 361/729, 361/733

FIELD-OF-SEARCH: 364/708.1, 361/680-686, 361/728-733, 395/281, 395/283, 395/500, 395/800

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

Search Selected | Search ALL

	PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
☐	5030128	July 1991	Herron et al.	439/372
☐	5123092	June 1992	Buxton et al.	395/250
☐	5265238	November 1993	Canova, Jr. et al.	395/500
☐	5323291	June 1994	Boyle et al.	361/683
☐	5377357	December 1994	Nishigaki et al.	395/800
☐	5430883	July 1995	Horiuchi	395/750
☐	5463742	October 1995	Kobayashi	395/281
☐	5473499	December 1995	Weir	361/58
☐	5526493	June 1996	Shu	395/281

ART-UNIT: 276

PRIMARY-EXAMINER: Moise; Emmanuel L.

ATTY-AGENT-FIRM: Dillon; Andrew J.

ABSTRACT:

A docking device for a portable computer includes a body adapted to receive a portable computer. The body encloses a number of electronic components adapted to be connected to a portable computer via a number of signal lines. A connector is provided in the body which mechanically connects the number of signal lines within the body to a corresponding number of signal lines within the portable computer. The docking device also includes a signal connection means interposed between the portable computer and the electronic components for selectively electrically interconnecting the particular ones of the signal lines within the body to corresponding signal lines among the number of signal lines within the portable computer in response to an operating state of said portable computer. A preferred embodiment of the docking device further includes a control means for controlling the selective electrical interconnection of the particular ones of the number of signal lines by the signal connection means. In this preferred embodiment, the control means electrically interconnects the particular ones of the number of signal lines within the body to corresponding signal lines within the portable computer in response to an acknowledgment by the portable computer of a request by the control means to electrically interconnect the particular ones of the number of signal lines.

17 Claims, 23 Drawing figures

WEST

End of Result Set

Generate Collection Print

L1: Entry 1 of 1 File: USPT Feb 2, 1999

DOCUMENT-IDENTIFIER: US 5867406 A
TITLE: Docking device for a portable computer and a method for docking a portable computer to the docking device

US Patent No. (1):
5867406

Detailed Description Text (8):
Finally, portable computer 50 includes an interrupt handler 24. The interrupt handler 24 continuously monitors the system bus 4 (more particularly, EVENT# of the bus signals) and detects the occurrence of a software interrupt also referred to as SMI (System Management Interrupt). In the depicted embodiment of the present invention, EVENT# in state ACTIVE LOW corresponds to a software interrupt in any device (including docking device 60) which notifies BIOS 73 of the interrupt, as described later.

Detailed Description Text (15):
Interface circuit 41 contains a plurality of registers such as an output register, input register, and control register. The output register is used to write the status or data to be reported to host 50 by docking device 60 and, more particularly, to the CPU 42. The input register is used to write the command or data to be sent to docking device 60 by host 50. The control register consists of plurality of bit flags such as IBF (Input Buffer Full), OBF (Output Buffer Full), and a busy flag. The IBF is set to 1 when host 50 writes a command or data to the input register and set to 0 when CPU 42 within docking device 60 reads the command or data in response to an interrupt generated by setting IBF. The OBF is set when CPU 42 writes a status or the like to the output register and reset when host 50 reads the output register in response to detecting the setting of the OBF when polling. The busy flag is a flag which indicates that CPU 42 is processing a task. While the busy flag is set, no command from host 50 is accepted. Further, the interface circuit 41 can issue an SMI to host 50 and CPU 42. Input to the interface circuit 41 is an eject signal for demanding the undocking of systems 50 and 60. The eject signal is generated when the user presses an eject button 44 disposed in the case of docking device 60 or by software executing within host 50.

Detailed Description Text (21):
Main DC power supply 86 is used to convert the DC current from the AC/DC converter 83 into a voltage and to supply the voltage to the various devices coupled to main bus 33, such as SCSI controller 35, PCMCIA controller 36, and IDE.sub.-- HDD 37. In this embodiment, in response to powering-on docked host 50, or the docking of host 50 while in a power-on state, which are detected by the signals DOCKED# and PWR.sub.-- ON#, respectively, main DC power supply 86 starts to supply power to the main bus 33. As will be appreciated by those skilled in the art, lines which transmit signals such as serial port, parallel port, keyboard, and mouse port signals are only an extension of the lines within host 50 and need not be supplied with power.

Detailed Description Text (28):
PWR.sub.-- ON#, SUS.sub.-- STAT#, DOCKED# and NOTE.sub.-- ID0 are provided for sending information from host 50 to docking device 60, while EVENT# is provided to send information from docking device 60 to host 50. If an event to be reported to

host 50 occurs within docking device 60, EVENT# becomes ACTIVE.sub.-- LOW. In addition, CPU 42 within docking device 60 writes the content of the event which has occurred to the output register in the interface circuit 41. The notification of the event by EVENT# is processed as a software interrupt (SMI) on host 50 side. More specifically, the interrupt handler 24 always monitors EVENT#, and reports the detection of ACTIVE.sub.-- LOW to BIOS 73. Then, BIOS 73 reads the output register in the interface circuit 41 in response to this notification to determine the content of the event. In particular, the event which docking device 60 wants to report to host 50 is a prior notification of docking (ABOUT.sub.-- TO.sub.-- DOCK) or a prior notification of undocking (ABOUT.sub.-- TO.sub.-- UNDOCK) (see sections D and E, below). Note that docking device 60 can utilize EVENT# on the assumption that host 50 is of the new type having the function of processing the SMI from docking device 60, as described above.

Detailed Description Text (31):
With reference now to FIG. 7, the process begins at block 100 when the connectors for connecting host 50 and docking device 60 are mechanically coupled by the user. Since the interface circuit 41 and the CPU 42 communicating with the subbus 34 within docking device 60 are already in an operating state, the mechanical connection can be detected by a change in DOCKED# to ACTIVE.sub.-- LOW as illustrated at step 102. Then, at block 104 it is determined from NOTE.sub.-- ID0 whether host 50 is a new type or an old type of portable computer. If host 50 is an old type, the SMI from the docking device 60 cannot be processed as described above, and thus system bus 4 cannot be electrically connected in the same operation. Accordingly, the process branches to off-page connector S.

Detailed Description Text (36):
Referring now to FIG. 8, if host 50 is in an ordinary operating state, the process proceeds from on-page connector P to block 200, which illustrates interface circuit 41 issuing an SMI to host 50 by changing EVENT# to ACTIVE.sub.-- LOW. Interrupt handler 24 detects the SMI and informs BIOS 73 of the SMI. Then, BIOS 73 searches for which device has sent the SMI and determines that docking device 60 is the source of the SMI. The process proceeds to block 202, which illustrates BIOS 73 reading ABOUT.sub.-- TO.sub.-- DOCK from the output register in the interface circuit 41 to determine if docking of host 50 is about to be performed and reporting the docking to OS 72. Also, at this point, main DC power supply 86 begins to supply power to devices coupled to main bus 33.

Detailed Description Text (41):
With reference now to FIG. 9, there is illustrated the continuation of the docking process following off-page connector Q of FIG. 7. As depicted, the process proceeds from on-page connector Q decision block 300, which illustrates the process awaiting a resume command. When a resume command is received, BIOS 73 within host 50 reads the content (`ABOUT.sub.-- TO.sub.-- DOCK`) of the output register in the interface circuit 41 and detects that a docking is about to be performed as depicted at block 302. A resume command can occur within host 50, such as a keyboard (function key) input, or can occur within docking device 60, such as the pressing of the power switch (not shown) of docking device 60. Block 302 is performed by resume code (usually loaded into the main memory 3 at initialization) within host 50 in the former case. In the latter case, resumption is triggered by the occurrence of the SMI and performed by BIOS 73. The process proceeds to block 204, which indicates connecting the main bus 33. Since blocks 204-218 were described above, the description thereof is omitted here.

Detailed Description Text (42):
Referring now to FIG. 10, there is depicted the continuation of the process after off-page connector R of FIG. 7. As illustrated, if host 50 is in a power-off state, the process loops at block 400 until the power supply is turned on. Then, when the power supply for the host 50 is turned on, a self-diagnostic program POST (Power-On Self-Test) is executed by host 50 at block 402. The POST code reads the content (`ABOUT.sub.-- TO.sub.-- DOCK`) of the output register in the interface circuit 41 to detect that a docking is about to be performed. In addition, docking device 60 detects the power-on of host 50 through PWR.sub.-- ON# and main DC power supply 86 starts to supply power to devices coupled to main bus 33.

Detailed Description Text (45):
With reference now to FIG. 11, the process begins at block 500 and proceeds to block 502, where a request for undocking is generated by the user pressing eject button 44 on the docking device 60 or by software executing within host 50. In the former case, CPU 42 is activated by the eject button 44 and, in the latter case, CPU 42 detects request for undocking by DOCKED#. Then, at block 504, it is determined by NOTE.sub.-- ID0 whether host 50 is a new type or an old type of portable computer. If host 50 is the old type, host 50 cannot process the SMI issued by docking device 60; thus, the process proceeds from block 504 to off-page connector W.

Detailed Description Text (46):
On the other hand, if a determination is made that host 50 is a new type, the process proceeds to block 506, where `ABOUT.sub.-- TO.sub.-- UNDOCK` is written to the output register in the interface circuit 41 to indicate that an undocking event has been detected. Then, the docking device 60 determines the power supply state of host 50 from PWR.sub.-- ON# and SUS.sub.-- STAT# at block 508. The process branches to off-page connector T when host 50 is in an ordinary operating state, to off-page connector U when host 50 is in a suspend state, and to off-page connector V when host 50 is in a power-off state.

Detailed Description Text (47):
Referring now to FIG. 12, there is depicted the undocking process subsequent to off-page connector T of FIG. 11. The process proceeds from on-page connector T to block 600, where if host 50 is in an ordinary operating state, the interface circuit 41 issues an SMI to host 50. The interrupt handler 24 detects and reports the occurrence of the SMI to BIOS 73. When BIOS 73 detects that the SMI has been issued by docking device 60, host 50 reads the content (`ABOUT.sub.-- TO.sub.-- UNDOCK`) of the output register in the interface circuit 41 and reports to OS 72 that host 50 and docking device 60 are about to be undocked at block 602. Then, OS 72 sends a command to docking device 60 to isolate the bus at block 604. More particularly, the command is transmitted by writing a record `BUS.sub.-- DISCONNECT` to the input register in the interface circuit 41 and setting the IBF. Setting the IBF causes an interrupt to occur in CPU 42. Thereafter, CPU 42 sets the busy flag to process `BUS.sub.-- DISCONNECT`. Then, the CPU 42 writes a code "Acknowledge" indicating acceptance of the command to the output register and sets the OBF. BIOS 73 polls the OBF and reads the content of the output register in response to setting the OBF. BIOS 72 confirms that the command has been accepted by resetting the OBF. Then, at block 606 CPU 42 determines through the resetting of the OBF that the command has been accepted by host 50 and causes the interface circuit 41 to isolate main bus 33 and subbus 34. Like connection of main bus 33, isolation is performed at the start of the refresh cycle. After the isolation of the bus signals 33 and 34, host 50 may be ejected from docking device 60 by a mechanical operation. Thereafter, the process terminates at block 608.

Detailed Description Text (48):
With reference now to FIG. 13, there is illustrated a flowchart of the undocking process following off-page connector U of FIG. 11. The process proceeds from on-page connector V to block 700, which depicts host 50 in a suspend state waiting for a resume command. When a resume command is given, the process proceeds to block 702, where BIOS 73 reads the content (`ABOUT.sub.-- TO.sub.-- UNDOCK`) of the output register in the interface circuit 41 to detect that undocking is about to be performed. A resume command can occur within host 50 or docking device 60, as described above in section D. Step 702 is executed by the resume code in the former case and by BIOS 73 in the latter case. Next, the process proceeds to blocks 704-708, which are substantially the same as blocks 604-608 described above. Therefore, the description thereof is omitted here.

Detailed Description Text (51):
If host 50 is a conventional portable computer, host 50 cannot handle the SMI from docking device 60, as has been described in section C. Consequently, those skilled in the art will appreciate that operations similar to sections D and E cannot be performed by a conventional portable computer. Thus, in a preferred embodiment of the present invention, if host 50 determined to be a conventional portable computer at block 104 of FIG. 7, the docking process continues following on-page connector S in FIG. 15. At block 900, docking devices 60 determines the power supply state of

host 50 from the states of PWR.sub.-- ON# and SUS.sub.-- STAT#. If docking is attempted when the host 50 is in an ordinary operating state, at block 902 the docking device 60 beeps to warn the user that docking is prohibited. More specifically, CPU 42 detects the attempted docking from NOTE.sub.-- ID0 of DOCKED# being high and activates an alarm (not shown). However, since main bus 33 is electrically isolated by main bus isolator 31, the hardware coupled to main bus 33 cannot be electrically damaged by a sudden docking. Accordingly, the power of host 50 need not be forcibly shut down as is done by some prior art system, enabling the user can avoid the suspension of a current task by such an inadvertent docking.

CLAIMS:

9. The docking device for a portable computer of claim 1, and further comprising:

a circuit that detects mechanical interconnection of said second plurality of signal lines and said connector.

11. A method for docking a portable computer to a docking device, said docking device having one or more electronic components that are electrically connectable to said portable computer via a first plurality of signal lines, and a connector within said body that is mechanically connectable to a second plurality of signal lines within said portable computer, said method comprising:

detecting mechanical connection of said connector provided in said body and said second plurality of signal lines within said portable computer;

in response to a detection of mechanical connection between said connector and said second plurality of signal lines, transmitting to said portable computer a request to electrically connect particular ones of said first plurality of signal lines within said body to corresponding signal lines among said second plurality of signal lines within said portable computer; and

in response to receipt of an acknowledgement from said portable computer of said request to electrically connect said particular ones of said first plurality of signal lines, electrically interconnecting said particular ones of said first plurality of signal lines within said body to corresponding signal lines among said second plurality of signal lines within said portable computer, such that electrical connection between said plurality of electronic components and said portable computer is achieved while said portable computer is in an ordinary operating state in which said second plurality of signal lines are active.

15. The method for docking a portable computer to a docking device of claim 11, said docking device further including a bus isolator coupled to said connector and a processor coupled to said bus isolator, said method further comprising the step of:

establishing electrical connection between said processor and said connector via said bus isolator in response to a detection of mechanical connection between said connector and said second plurality of signal lines.